IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent No.: **6,856,975**

Application No.: **09/540,193**

Applicant: **Frank Inglis** Group Art Unit: **3624**

Assignee: **Verify & Protect, Inc.** Examiner: **Weisberger, Richard C**

Filed: **March 30, 2000**

Granted: **February 15, 2005**

For: **SYSTEM, METHOD, AND ARTICLE OF MANUFACTURE FOR SECURE TRANSACTIONS UTILIZING A COMPUTER NETWORK**

FILED ELECTRONICALLY

United States Patent and Trademark Office
Customer Service Window, Mail Stop Petitions
Randolph Building
401 Dulany Street
Alexandria, VA 22314

PETITION UNDER 37 C.F.R. 1.378(b) FOR ACCEPTANCE OF UNAVOIDABLY DELAYED MAINTENANCE FEE PAYMENT

I. Introduction.

Petitioner Verify and Protect, Inc. respectfully requests revival of U.S. Patent No. 6,856,975 (the "'975 patent") and acceptance of the enclosed maintenance fee pursuant to 37 C.F.R. 1.378(b). As explained in the attached Declarations of Michael Lang and Jeffry Boston, Petitioner took steps to ensure timely payment of fees and acted promptly once it was discovered that payment had been delayed. Petitioner meets the reasonably prudent person standard set forth by the U.S. Patent & Trademark Office ("PTO") and as further articulated by the U.S.

Court of Appeals for the Federal Circuit and is therefore entitled to have the unavoidably delayed maintenance fee accepted.

II. Background.

In 2000, Mr. Michael Lang was an officer of Verify and Protect, Inc. ("VPI"), a technology company. VPI retained the Halvorson Law Firm to prepare a patent application, prosecute the application, and monitor and calendar relevant due dates in the Halvorson Law Firm docketing system. (*See* Declaration of Michael Lang, attached as Exhibit A, at paras. 3-5). In March 2000, employee and inventor, Mr. Frank Inglis, through the Halvorson Law Firm, filed United States Patent Application No. 09/540,193 (the "'193 application"), entitled "A System, Method and Article of Manufacture for Securing Transactions Utilizing a Computer Network," which was assigned to VPI. (*See* Exhibit A at para. 3; *see also* Exhibit A-1). VPI assisted the Halvorson Law Firm during the prosecution of the '193 application by formulating arguments in response to Office Actions and had frequent contact with the Halvorson Law Firm during the prosecution of the '193 application. (*See* Declaration of Jeffry Boston, attached as Exhibit B, at para. 4).

On February 15, 2005, the '193 application issued as the '975 patent. (*See* Exhibit A-2). VPI understood that maintenance fees were to be paid at specific intervals as required by the PTO and to ensure timely payment, VPI relied on the Halvorson Law Firm to correctly calendar the maintenance fee due dates in their docketing system. (*See* Exhibit A at para. 7). VPI also expected the Halvorson Law Firm to provide notice to VPI when payments were due. (*See id.* at para. 7). The Halvorson Law Firm duly updated the firm's contact information when paying the Issue Fee on behalf of VPI in order to ensure that the PTO could properly maintain contact with the firm for the '975 patent. (*See* Exhibit A-3).

Later in 2005, VPI contemplated a business transaction with a company called IntelAgents, Inc. Mike Lang had a role in the negotiations as one of VPI's managing executives. The parties drafted an Agreement (the "2005 Agreement") that required VPI to transfer all of its assets (including ownership of the '975 patent) to IntelAgents, Inc. (*See* Exhibit A-4; *see also* Exhibit A at para. 8). IntelAgents, Inc. later changed its name to InfrAegis, Inc. (*See* Exhibit A-6; *see also* Exhibit A at para. 9).

In April 2008, InfrAegis and VPI agreed that certain conditions precedent to the 2005 Agreement were never met and that the transfer of assets never actually occurred. (*See* Exhibit A at paras. 10-11). The parties executed a General and Mutual Release (the "2008 Release") to further document their understanding. (*See* Exhibit A-7).

Since the issuance of the '975 patent in 2005, VPI continued to rely on the Halvorson Law Firm and its docketing system to provide notice of payments and actions required by the PTO. (*See* Exhibit A at para. 12).

On February 15, 2008, a six-month window for paying the four-year maintenance fee began, but VPI did not receive any notice from the Halvorson Law Firm before, during, or after the six-month window that a payment was due. (*See* Exhibit A at paras. 13-18; Exhibit B at paras. 9-14). On August 15, 2008, a six month window for payment of the maintenance fee with a surcharge began, and VPI still did not receive any notification before, during, or after this six month window that a payment was due by February 15, 2009. (*See id.*). The Halvorson Law Firm did not pay the maintenance fee. Ultimately, the '975 patent became expired. Even after expiration, the Halvorson Law Firm, who VPI relied upon as its registered agent and primary contact with the PTO, did not send a notice regarding the failure to pay the maintenance fee or the expiration of the '975 patent. (*See* Exhibit B at paras. 10-12).

VPI first realized that the '975 patent had been expired in November 2011 and took immediate steps to revive it by hiring a different law firm. (*See* Exhibit B at para. 13). All actions taken by VPI to avoid abandonment of the '975 patent have been reasonable and calculated to avoid the delay or non-payment of fees. (*See id.* at para. 14). VPI's delay in payment of the four-year maintenance fee was therefore unavoidable due to the reliance of VPI on its patent counsel at the Halvorson Law Firm.

III. Legal Standard.

Decisions on reinstating a lapsed patent are made by applying the "reasonably prudent person standard." *Ray v. Lehman*, 55 F.3d 606, 608-09 (Fed. Cir. 1995). Courts have interpreted the "unavoidable delay" standard as requiring the individual or entity in question to have exercised the due care and diligence "generally used and observed by prudent and careful men in relation to their most important business." *R.R. Donnelley & Sons Co. v. Dickinson*, 123 F.Supp.2d 456, 459 (N.D. Ill. 2000). Whether or not delay is unavoidable is decided on a case-by-case basis, taking all of the facts and circumstances into account. *See Smith v. Mossinghoff*, 671 F.2d 533, 538 (D.C. Cir. 1982).

IV. Argument.

VPI retained the Halvorson Law Firm as patent counsel to obtain the '975 patent. (Exhibit A at para. 3). The arrangement with the Halvorson Law Firm was designed to ensure VPI that due dates related to the '975 patent were monitored by the firm and its docketing system. (Exhibit A at para. 5). In 2005, when the patent issued, VPI continued to rely upon the Halvorson Firm and expected them to continue their duty by maintaining its records and notifying VPI when a payment became due. Such reliance on the skill and attention of authorized legal counsel is the definition of the due care and diligence generally used and

observed by careful individuals in relation to their most important business. *R.R. Donnelley & Sons*, 123 F.Supp.2d at 459. VPI acted promptly upon discovery that the '975 patent was expired. In November 2011, when Michael Lang and Jeffry Boston became aware that the patent had expired for failure to pay the four-year maintenance fee, VPI hired new patent counsel and filed this petition within weeks after the reinstatement of the corporate entity. The delayed maintenance fee payment is thus completely attributable to previous patent counsel's failure to provide appropriate notice or to effect payment on VPI's behalf.

V. Conclusion.

In light of the foregoing facts and arguments, Petitioner respectfully requests that the PTO grant this petition, revive the '975 patent, and accept the four-year maintenance fee as unavoidably delayed.

Respectfully submitted,

Dated: April 23, 2012

By: /Eric Sophir, Reg. No. 48,499/
Eric Sophir
Registration No. 48,499
SNR Denton US LLP
1301 K Street, NW
Suite 600, East Tower
Washington, DC 20005
(202) 408-6470

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent No.: **6,856,975**	OFFICE OF PETITIONS
Applicant: **Frank Inglis**	Title: **SYSTEM, METHOD, AND ARTICLE OF MANUFACTURE FOR SECURE TRANSACTIONS UTILIZING A COMPUTER NETWORK**
Assignee: **Verify & Protect, Inc.**	
Filed: **March 30, 2000**	
Granted: **February 15, 2005**	

April 12, 2012

FILED ELECTRONICALLY FILED VIA EFS-WEB

Declaration of Michael Lang

1. I, Michael Lang, hereby state and affirm the following:

2. I am Chairman of Verify and Protect, Inc. ("VPI"), the assignee of all of the rights and interest to U.S. Patent No. 6,856,975 (the "'975 patent"). (copy of Assignment attached as Exhibit 1).

3. In March of 2000, the Halvorson Law Firm was retained to represent inventor Frank Inglis, and after assignment, VPI, before the United States Patent & Trademark Office ("PTO") and to acquire patent rights to a "System Method and Article of Manufacture for Secure Transactions Utilizing a Computer Network."

4. On March 30, 2000 the Halvorson Law Firm, working at the request of VPI and Mr. Inglis, filed U.S. Patent Application Serial No. 09/540,193 (the "'193 application").

5. As a part of our agreement with the Halvorson Law Firm, VPI established that all PTO due dates related to both the prosecution and maintenance of the '193 application and resulting patent would be managed by the Halvorson Law Firm and their docketing system

6. On February 15, 2005, the '193 application issued as the '975 patent. (copy of '975 patent attached as Exhibit 2)

7. I understood, on behalf of VPI, that certain fees were to be paid in connection with the patent application and issued patent as required by the PTO. To ensure timely payment of all fees, VPI relied on the Halvorson Law Firm to correctly calendar the maintenance fee due dates on their docketing system and provide notice when payments were due. The Halvorson Law Firm took steps to maintain current contact information at the PTO as attorneys of record for the '975 patent. (Copy of docket sheet showing that the Halvorson Law Firm updated and confirmed its contact address when the issue fee was paid attached as Exhibit 3).

8. In 2005, VPI contemplated a business transaction with a company called IntelAgents, Inc. The parties drafted an Agreement (the "2005 Agreement") that called for VPI to be merged into and to transfer all of its assets to an acquisition entity owned and controlled by IntelAgents, Inc. . (copy of 2005 Agreement attached as Exhibit 4) Thereafter, I transferred the books and records for the operation of VPI to the managers of IntelAgents, and I relied upon IntelAgents to manage the assets of VPI and to pay any fees necessary to maintain VPI's patent rights.

9. In 2005, IntelAgents, Inc. changed its name to InfrAegis, Inc. (copy of Illinois Secretary of State report attached as Exhibit 5).

10. Infraegis failed to pay the consideration required under the 2005 Agreement or to otherwise take the necessary steps to conclude the merger and transfer of assets to InfrAegis.

11. In 2008, InfrAegis, Inc. and VPI executed a General and Mutual Release (the "2008 Release") in which the parties agreed that certain terms of the 2005 Agreement were never met and that the transfer of assets never occurred, thus unwinding the transaction contemplated by the 2005 Agreement and returning the assets to VPI, including the '975 patent. (copy of the 2008 Release attached as Exhibit 6) Thus, VPI reacquired control of the assets in 2008. The principal asset of VPI as of 2008 was the '975 patent. The inventor, Mr. Inglis, continues to have a monetary interest in VPI and VPI's rights to the invention of the '975 patent that he assigned to VPI.

12. During the time period described above, and at all times since the filing of the '193 application, VPI continued to rely on the Halvorson Law Firm and its docketing system to provide notice of deadlines related to the '975 patent.

13. In November 2011, VPI began to take steps to develop, implement, and practice the invention of the '975 patent. In connection with those efforts, we engaged counsel to advise on intellectual property rights.

14. After VPI engaged new counsel, I learned in November 2011 that the four-year maintenance fee payment for the '975 patent had become due but had not been paid. I have since learned that the four-year maintenance fee payment for the '975 patent was due by February 15, 2009.

15. VPI was not aware of the due date for the Maintenance Fee and was relying upon its patent counsel, the Halvorson Law Firm, to notify VPI of the fees that were due to the PTO in connection with preserving the patent rights for the '975 patent. VPI never received any form of

notice from The Halvorson Law Firm, or any other source, that a maintenance fee payment was due.

16. As of February 15, 2009, VPI had not received any notice from the Halvorson Law Firm, or any other source, that a maintenance fee payment was due.

17. VPI did not receive a notice of abandonment from the PTO or from the Halvorson Law Firm.

18. VPI first realized that the '975 patent had been abandoned in November 2011 and took prompt steps to revive it by retaining a different law firm to protect its rights to the '975 patent.

19. All actions taken by VPI with respect to the '975 patent have been reasonable and calculated to avoid non-payment of fees.

20. Non-payment of the four-year maintenance fee was unavoidable due to the reliance of VPI on its patent counsel at the Halvorson Law Firm.

22. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true and further that the statements are made with the knowledge that willful statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements jeopardize the validity of the application or any patent issuing thereon.

April 2, 2012

Michael Lang
Chairman, Verify and Protect, Inc.

EXHIBIT A

ASSIGNMENT BY INVENTOR OF PATENTS

For good and valuable consideration, the receipt of which is hereby acknowledged, the below signed inventor, a citizen of the United States of America, residing at the addresses listed below his name, sells and assigns to Verify and Protect, Inc., a Delaware Corporation having a place of business at 3333 Warrenville Road, Suite 200, Lisle IL 60532, its successors and assigns, all their right, title and interest in and to the following patents and patent application:

(1) U.S. Patent Application, Currently Entitled "System, Method, and Article of Manufacture for Secure Transactions Utilizing a Computer Network", Ser. No. 09/540,193, filed on 03/30/2000.

and all inventions contained therein, all improvements thereon, all technologies related thereto, all reissues and extensions thereof, and covenant that he has full right so to do, and agrees that he will communicate to said Corporation or its representatives any facts known to him respecting said improvements and testify in any legal proceeding, sign all lawful papers, execute all, reissue and extension applications, make all rightful oaths, and generally do everything possible to aid said Corporation, its successors, assigns and nominees, to obtain and enforce patent protection for said inventions in all countries.

IN TESTIMONY WHEREOF, I hereunto set my hand and seal this 15th day of OCT, 2002.

Frank H. Inglis
3215 E. Muirwood Dr.
Phoenix, Arizona 85044

State of Arizona ss. County of Maricopa

On this 15th day of Oct, 2002 before me, a Notary Public in and for the State and County aforesaid, personally appeared and to me known be the person of the above signed names, who signed and sealed the foregoing instrument, and they acknowledged the same to be their free act and deed.

Lisa M. Biondi
Notary Public

(Seal)

OFFICIAL SEAL
LISA M. BIONDI
NOTARY PUBLIC-ARIZONA
MARICOPA COUNTY
MY COMM. EXPIRES DEC. 13, 2005

US006856975B1

(12) **United States Patent**
Inglis

(10) **Patent No.: US 6,856,975 B1**
(45) **Date of Patent: Feb. 15, 2005**

(54) **SYSTEM, METHOD, AND ARTICLE OF MANUFACTURE FOR SECURE TRANSACTIONS UTILIZING A COMPUTER NETWORK**

(75) Inventor: **Frank Inglis**, Phoenix, AZ (US)

(73) Assignee: **Verify & Protect Inc.**, Lisle, IL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/540,193**

(22) Filed: **Mar. 30, 2000**

(51) **Int. Cl.**[7] **G06F 17/00**

(52) **U.S. Cl.** **705/51**; 705/1; 705/40; 705/54

(58) **Field of Search** 705/51, 54, 1

(56) **References Cited**

PUBLICATIONS

Spring, E-Business Security Technologies, 2001.*

* cited by examiner

Primary Examiner—Richard Weisberger
(74) *Attorney, Agent, or Firm*—The Halvorson Law Firm

(57) **ABSTRACT**

The present invention is a system or method and device useful for the secure electronic payment of consumer debts over a publicly accessible computer network. The preferred form of the present invention uses at least two separate, but compatible, software packages. Security server software that continuously runs on a security server and payor software that runs on demand on a payor computer system. The payor computer system communicates via the payor software with the security server via the security server software. The communication, or transaction, session operates under the secure communication protocol described below. A payee computer system may also communicate via payee software with the security server. Additionally, a version is provided that utilizes smart card technology and a remote kiosk computer that communicates with the security server.

10 Claims, 6 Drawing Sheets

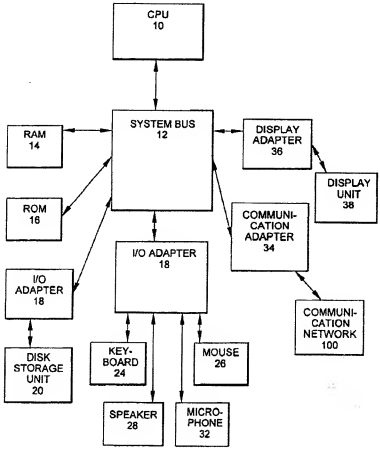


Fig. 1

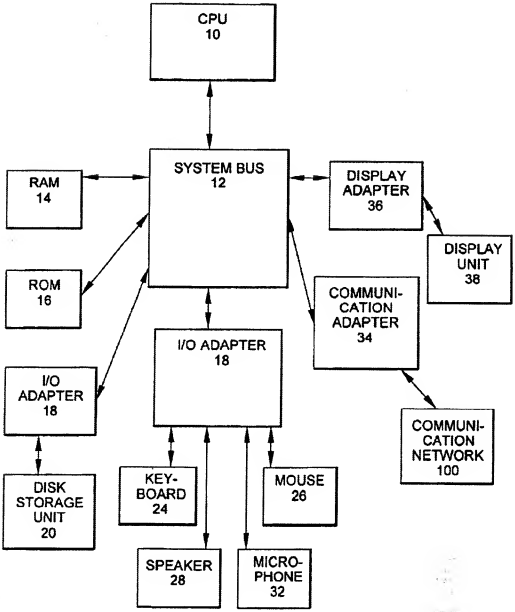

CPU
10
SYSTEM BUS
12
RAM
14
ROM
16
I/O
ADAPTER
18
I/O ADAPTER
18
DISK
STORAGE
UNIT
20
DISPLAY
ADAPTER
36
DISPLAY
UNIT
38
COMMUNI-
CATION
ADAPTER
34
COMMUNI-
CATION
NETWORK
100
KEY-
BOARD
24
MOUSE
26
SPEAKER
28
MICRO-
PHONE
32

Fig. 2

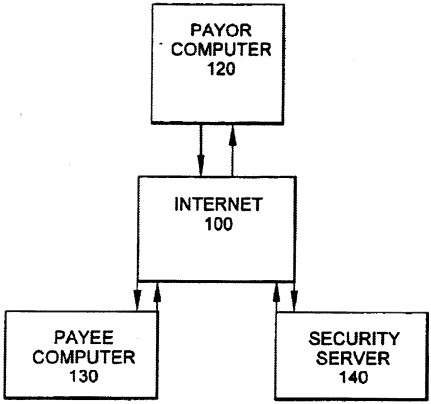

PAYOR
COMPUTER
120
INTERNET
100
PAYEE
COMPUTER
130
SECURITY
SERVER
140

Fig. 3A

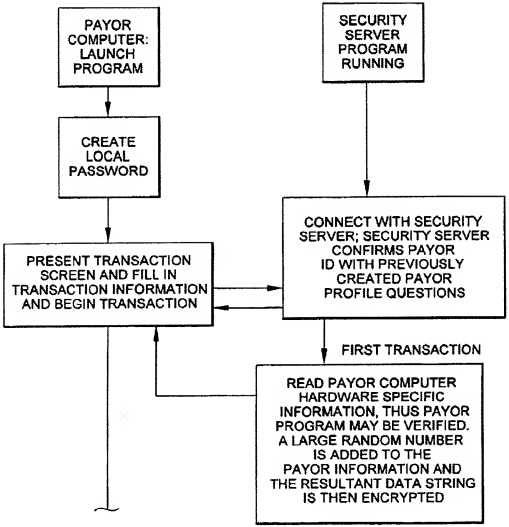

PAYOR COMPUTER: LAUNCH PROGRAM
SECURITY SERVER PROGRAM RUNNING
CREATE LOCAL PASSWORD
CONNECT WITH SECURITY SERVER; SECURITY SERVER CONFIRMS PAYOR ID WITH PREVIOUSLY CREATED PAYOR PROFILE QUESTIONS
PRESENT TRANSACTION SCREEN AND FILL IN TRANSACTION INFORMATION AND BEGIN TRANSACTION
FIRST TRANSACTION
READ PAYOR COMPUTER HARDWARE SPECIFIC INFORMATION, THUS PAYOR PROGRAM MAY BE VERIFIED. A LARGE RANDOM NUMBER IS ADDED TO THE PAYOR INFORMATION AND THE RESULTANT DATA STRING IS THEN ENCRYPTED

Fig. 3B

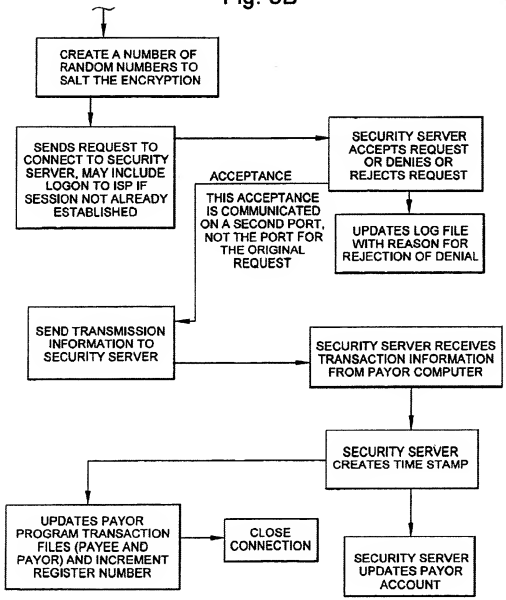

CREATE A NUMBER OF RANDOM NUMBERS TO SALT THE ENCRYPTION
SENDS REQUEST TO CONNECT TO SECURITY SERVER, MAY INCLUDE LOGON TO ISP IF SESSION NOT ALREADY ESTABLISHED
SECURITY SERVER ACCEPTS REQUEST OR DENIES OR REJECTS REQUEST
ACCEPTANCE
THIS ACCEPTANCE IS COMMUNICATED ON A SECOND PORT, NOT THE PORT FOR THE ORIGINAL REQUEST
UPDATES LOG FILE WITH REASON FOR REJECTION OF DENIAL
SEND TRANSMISSION INFORMATION TO SECURITY SERVER
SECURITY SERVER RECEIVES TRANSACTION INFORMATION FROM PAYOR COMPUTER
SECURITY SERVER CREATES TIME STAMP
UPDATES PAYOR PROGRAM TRANSACTION FILES (PAYEE AND PAYOR) AND INCREMENT REGISTER NUMBER
CLOSE CONNECTION
SECURITY SERVER UPDATES PAYOR ACCOUNT

Fig. 4A

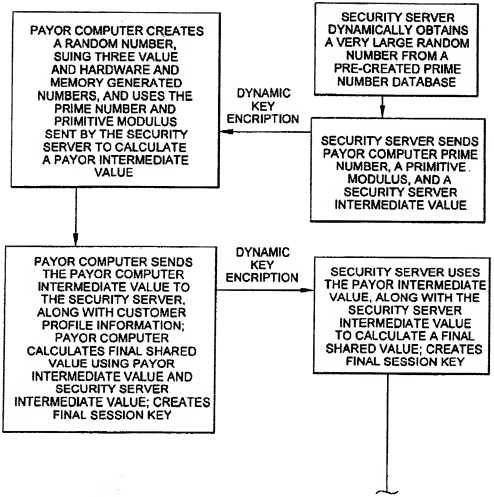
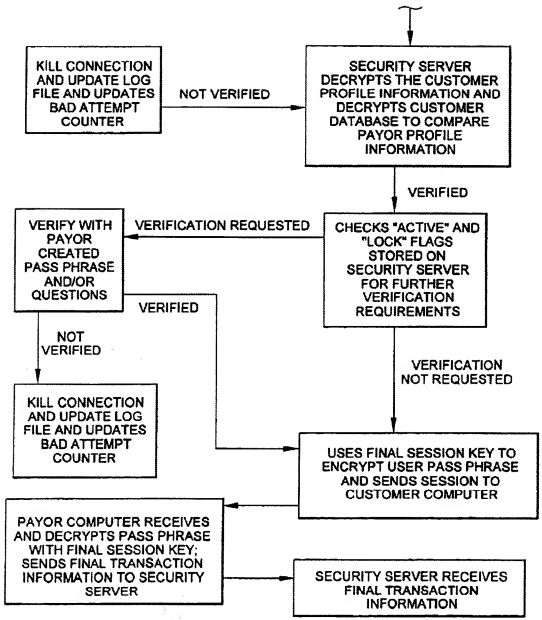

PAYOR COMPUTER CREATES A RANDOM NUMBER, SUING THREE VALUE AND HARDWARE AND MEMORY GENERATED NUMBERS, AND USES THE PRIME NUMBER AND PRIMITIVE MODULUS SENT BY THE SECURITY SERVER TO CALCULATE A PAYOR INTERMEDIATE VALUE
SECURITY SERVER DYNAMICALLY OBTAINS A VERY LARGE RANDOM NUMBER FROM A PRE-CREATED PRIME NUMBER DATABASE
DYNAMIC KEY ENCRIPTION
SECURITY SERVER SENDS PAYOR COMPUTER PRIME NUMBER, A PRIMITIVE MODULUS, AND A SECURITY SERVER INTERMEDIATE VALUE
PAYOR COMPUTER SENDS THE PAYOR COMPUTER INTERMEDIATE VALUE TO THE SECURITY SERVER, ALONG WITH CUSTOMER PROFILE INFORMATION; PAYOR COMPUTER CALCULATES FINAL SHARED VALUE USING PAYOR INTERMEDIATE VALUE AND SECURITY SERVER INTERMEDIATE VALUE; CREATES FINAL SESSION KEY
DYNAMIC KEY ENCRIPTION
SECURITY SERVER USES THE PAYOR INTERMEDIATE VALUE, ALONG WITH THE SECURITY SERVER INTERMEDIATE VALUE TO CALCULATE A FINAL SHARED VALUE; CREATES FINAL SESSION KEY

Fig. 4B


KILL CONNECTION AND UPDATE LOG FILE AND UPDATES BAD ATTEMPT COUNTER
NOT VERIFIED
SECURITY SERVER DECRYPTS THE CUSTOMER PROFILE INFORMATION AND DECRYPTS CUSTOMER DATABASE TO COMPARE PAYOR PROFILE INFORMATION
VERIFIED
VERIFY WITH PAYOR CREATED PASS PHRASE AND/OR QUESTIONS
VERIFICATION REQUESTED
CHECKS "ACTIVE" AND "LOCK" FLAGS STORED ON SECURITY SERVER FOR FURTHER VERIFICATION REQUIREMENTS
VERIFIED
NOT VERIFIED
VERIFICATION NOT REQUESTED
KILL CONNECTION AND UPDATE LOG FILE AND UPDATES BAD ATTEMPT COUNTER
USES FINAL SESSION KEY TO ENCRYPT USER PASS PHRASE AND SENDS SESSION TO CUSTOMER COMPUTER
PAYOR COMPUTER RECEIVES AND DECRYPTS PASS PHRASE WITH FINAL SESSION KEY; SENDS FINAL TRANSACTION INFORMATION TO SECURITY SERVER
SECURITY SERVER RECEIVES FINAL TRANSACTION INFORMATION

SYSTEM, METHOD, AND ARTICLE OF MANUFACTURE FOR SECURE TRANSACTIONS UTILIZING A COMPUTER NETWORK

FIELD OF THE INVENTION

The present invention relates to the secure, electronic payment of consumer debt over a communication network, and more specifically, to a system, method and article of manufacture for securely transmitting payment information from a payor to a security server, which processes the transaction, and returning a confirmation of said payment.

BACKGROUND

The present invention relates to a method, device utilizing an electronic graphical representation of a monetary system for implementing electronic money payments as an alternative medium of economic exchange to cash, checks, credit and debit cards, and traditional electronic funds transfer. The system according to the present invention utilizes electronic representations of money that are designed to be universally accepted and exchanged as economic value by subscribers of the monetary system.

Currently, approximately 350 billion monetary transactions occur between individuals and institutions annually. The extensive use of monetary transactions has limited the automation of individual transactions such as purchases, fares, and bank account deposits and withdrawals. Individual cash transactions are burdened by the need to have the correct amount of cash or providing change therefor. Furthermore, the handling and managing of paper cash and coins is inconvenient, costly, and time consuming for both individuals and financial institutions.

Although checks may be written for any specific amount up to the amount available in the account, checks have very limited transferability and must be supplied from a physical inventory. Paper-based checking systems do not offer sufficient relief from the limitations of cash transactions, sharing many of the inconveniences of handling currency while adding the inherent delays associated with processing checks. To this end, economic exchange is moving toward automation for greater convenience at a lower cost.

Automation is being used for large transactions through computerized electronic funds transfer ("EFT") systems. ETF is essentially a process of value exchange achieved through the banking system's centralized computer transactions. EFT services are a transfer of payments utilizing electronic "letters of credit" and are used primarily by large commercial organizations. The American Clearing House (ACH), where a user can enter a pre-authorized code and download information with billing occurring later, and Point Of Sale (POS) systems, where transactions are processed by connecting with a central computer for authorization for the transaction granted or denied immediately, are examples of EFT systems that are utilized by retail and commercial organizations.

Home banking bill payment services are another example of EFT systems used by individuals to make payments from a home computer. Currently, however, home banking initiatives have found few payors. Less than one percent of bank payors use service accounts for transfers and information, using personal computers over telephone lines. One reason that home banking has not been a successful product is because the payor cannot deposit and withdraw money as needed in this type of system. Another reason home banking initiatives have found few payors is the inherent distrust in the security of data transmission of financial data across the Internet **100** prevalent in society given the present Internet **100** security and encryption products currently available to the general public.

Current EFT systems, credit cards, or debit cards, which are used in conjunction with an online system to transfer money between accounts, such as between the account of a merchant and that of a payor, do not satisfy the need for an automated transaction system providing an ergonomic interface.

To implement an automated, convenient transaction that can dispense some form of economic value, there has been a trend towards off-line payments. For example, numerous ideas have been proposed for some form of "electronic money" that can be used in non-cash payment transactions as alternatives to the traditional currency and check types of payment systems. Best known of these are magnetic stripe cards purchased for a given amount and from which a prepaid value can be deducted for specific purposes. Upon exhaustion of the economic value, the cards may be thrown away. Other examples include memory cards or so called smart cards, which are capable of repetitively storing information representing value that is likewise deducted for specific purposes. These methods also do not satisfy the current needs for a consumer friendly, convenient and secure electronic transaction system.

The Internet has become a valuable tool for the electronic transfer of information, which can include financial transactions. It is possible and desirable for a computer operating under the control of the payor over a publicly accessible packet-switched network (e.g., the Internet) to bi-directionally share payment information with a computer operated under the control of a payee, without risking the exposure of the information to interception by third parties that have access to the network, and to assure that the information is from an authentic source. It is further desirable for this information, including a subset of the information provided by the payor, to be provided to the payee by the security server system that is designated by a bank or other financial institution that has the responsibility of providing payment on behalf of the payor, without the risk of exposing that information to interception by third parties. Such institutions may include, for example, merchants or financial institutions.

One such attempt to provide such a secure transmission channel is a secure payment technology such as Secure Electronic Transaction (hereinafter "SET"), jointly developed by the Visa and MasterCard card associations, and described in Visa and MasterCard's Secure Electronic Transaction (SET) Specification, Feb. 23, 1996, hereby incorporated by reference. Other such secure payment technologies include Secure Transaction Technology ("STT"), Secure Electronic Payments Protocol ("SEPP"), Internet Keyed Payments ("IKP"), Net Trust, and Cybercash Credit Payment Protocol. One of ordinary skill in the art readily comprehends that any of the secure payment technologies can be substituted for the SET protocol without undue experimentation.

Such secure payment technologies, referenced above, require the payor to operate software that is compliant with the secure payment technology, interacting with third-party certification authorities, thereby allowing the payor to transmit encoded information to a payee, some of which may be decoded by the payee, and some which can be decoded only by an institution specified by the payor.

Another such attempt to provide such a secure transmission channel is a general-purpose secure communication protocol such the Secure Sockets Layer (hereinafter "SSL"). SSL provides a means for secure transmission between two computers. SSL has the advantage that it does not require special-purpose software to be installed on the payor's computer because it is already incorporated into widely available software that many people utilize as their standard Internet access medium. Other examples of general-purpose secure communication protocols include Private Communications Technology ("PCT") from Microsoft, Inc.; Secure Hyper-Text Transport Protocol ("SHTTP") from Terisa Systems; Shen; Kerberos; Photuris; and Pretty Good Privacy ("PGP") all of which meet the IPSEC criteria. One of ordinary skill in the art readily comprehends that any of the general-purpose secure communication protocols can be substituted for the SSL transmission protocol without undue experimentation. However these protocols have proven to be vulnerable to attack, therefore greater security must be available.

Banks desire an Internet payment solution that functions similar to existing Point of Sale (POS) applications that are currently installed on their host computers and require minimal changes to their host systems. This is a critical requirement since any downtime for a bank's host computer system represents an enormous expense. Currently, there are over fourteen hundred different payment-related applications available. The large number of applications is necessary to accommodate a wide variety of host message formats, diverse methods for communicating to a variety of hosts with different dial-up and direct-connect schemes, and different certification around the world.

Internet-based payment solutions require additional security measures that are not found in conventional POS or EFT terminals. This additional requirement is necessitated because Internet communication is done over publicly accessible unsecured communication line in stark contrast to the private, secure, dedicated phone or leased line service utilized between a traditional payee and an acquiring bank. Thus, it is critical that any solution utilizing the Internet for a communication backbone employs some form of secure cryptography.

As discussed above, the current state-of-the-art in Internet based payment processing is a protocol referred to as SET, or Secure Electronic Transaction. Since the SET messages are uniform across all implementations, banks cannot differentiate themselves in any reasonable way. Also, since SET is not a proper superset of all protocols utilized today, there are bank protocols that cannot be mapped or translated into SET because they require data elements for which SET has no placeholder. Further, SET only handles the message types directly related to authorizing and capturing credit card transactions and adjustments to these authorizations or captures. In a typical EFT terminal in the physical world, these messages comprise almost the entire volume of the total number of messages between the payee and the authorizing bank, but only half of the total number of different message types. These message types, which are used infrequently, but which are critical to the operation of the EFT terminal must be supported for proper transaction processing.

Generally, applications written for this field are written using JAVA, C, and/or the C++ languages and utilize object oriented programming methodology. Object oriented programming (OOP) has become increasingly used to develop complex applications. As OOP moves toward the mainstream of software design and development, various software solutions require adaptation to make use of the benefits of OOP.

OOP is a process of developing computer software using objects, including the steps of analyzing the problem, designing the system, and constructing the program. An object is a software package that contains both data and a collection of related structures and procedures. Since it contains both data and a collection of structures and procedures, it can be visualized as a self-sufficient component that does not require other additional structures, procedures or data to perform its specific task. OOP, therefore, views a computer program as a collection of largely autonomous components, called objects, each of which is responsible for a specific task. This concept of packaging data, structures, and procedures together in one component or module is called encapsulation.

In general, OOP components are reusable software modules that present an interface that conforms to an object model and that are accessed at run-time through component integration architecture. Component integration architecture is a set of architecture mechanisms that allow software modules in different process spaces to utilize each other's capabilities or functions. This is generally done by assuming a common component object model on which to build the architecture.

It is worthwhile to differentiate between an object and a class of objects at this point. An object is a single instance of the class of objects, which is often just called a class. A class of objects can be viewed as a blueprint, from which many objects can be formed.

OOP allows the programmer to create an object that is a part of another object. For example, the object representing a piston engine is said to have a composition-relationship with the object representing a piston. In reality, a piston engine comprises a piston, valves and many other components; the fact that a piston is an element of a piston engine can be logically and semantically represented in OOP by two objects.

OOP also allows creation of an object that "depends on" another object. If there are two objects, one representing a piston engine and the other representing a piston engine wherein the piston is made of ceramic, then the relationship between the two objects is not that of composition. A ceramic piston engine does not make up a piston engine. Rather it is merely one kind of piston engine that has one more limitation than the piston engine; its piston is made of ceramic. In this case, the object representing the ceramic piston engine is called a derived object, and it inherits all of the aspects of the object representing the piston engine and adds further limitation or detail to it. The object representing the ceramic piston engine "depends from" the object representing the piston engine. The relationship between these objects is called inheritance.

When the object or class representing the ceramic piston engine inherits all of the aspects of the objects representing the piston engine, it inherits the thermal characteristics of a standard piston defined in the piston engine class. However, the ceramic piston engine object overrides these ceramic specific thermal characteristics, which are typically different from those associated with a metal piston. It skips over the original and uses new functions related to ceramic pistons. Different kinds of piston engines have different characteristics, but may have the same underlying functions associated with it (e.g., how many pistons in the engine, ignition sequences, lubrication, etc.). To access each of these functions in any piston engine object, a programmer would call the same functions with the same names, but each type of piston engine may have different/overriding implemen-

tations of functions behind the same name. This ability to hide different implementations of a function behind the same name is called polymorphism and it greatly simplifies communication among objects.

With the concepts of composition-relationship, encapsulation, inheritance and polymorphism, an object can represent just about anything in the real world. In fact, our logical perception of the reality is the only limit on determining the kinds of things that can become objects in object-oriented software. Some typical categories are illustrated as follows: objects can represent physical objects, such as automobiles in a traffic-flow simulation, electrical components in a circuit-design program, financial transactions in an economics model, or aircraft in an air-traffic-control system; objects can represent elements of the computer-user environment such as windows, menus or graphics objects; an object can represent an inventory, such as a personnel file or a table of the latitudes and longitudes of cities; or an object can represent user-defined data types such as time, angles, and complex numbers, or points on the plane.

Programming languages are beginning to fully support the OOP principles, such as encapsulation, inheritance, polymorphism, and composition-relationship. With the advent of the C++ language, many commercial software developers have embraced OOP. C++ is an OOP language that offers a fast, machinc-executable code. Furthermore, C++ is suitable for both commercial-application and systems-programming projects. For now, C++ appears to be the most popular choice among many OOP programmers, but there is a host of other OOP languages, such as Smalltalk, common lisp object system (CLOS), and Eiffel. Additionally, OOP capabilities are being added to more traditional popular computer programming languages such as Pascal.

The development of graphical user interfaces began to turn procedural programming arrangements inside out. These interfaces allow the user, rather than program logic, to drive the program and decide when certain actions should be performed. Today, most personal computer software accomplishes this by means of an event loop that monitors the mouse, keyboard, and other sources of external events and calls the appropriate parts of the programmer's code according to actions that the user performs. The programmer no longer determines the order in which events occur. Instead, a program is divided into separate pieces that are called at unpredictable times and in an unpredictable order. By relinquishing control in this way to users, the developer creates a program that is much easier to use. Nevertheless, individual pieces of the program written by the developer still call libraries or objects provided by the operating system to accomplish certain tasks, and the programmer must still determine the flow of control within each piece after it's called by the event loop. Application code still "sits on top of" the system.

Even event loop programs requirc programmers to write a lot of code that should not need to be written separately for every application. The concept of an application framework carries the event loop concept further. Instead of dealing with all the nuts and bolts of constructing basic menus, windows, and dialog boxes and then making these things all work together, programmers using application frameworks start with working application code and basic user interface elements in place. Subsequently, they build from there by replacing some of the generic capabilities of the framework with the specific capabilities of the intended application.

Application frameworks reduce the total amount of code that a programmer has to write from scratch. However, because the framework is really a generic application that displays windows, supports copy and paste, and so on, the programmer can also relinquish control to a greater degree than event loop programs permit. The framework code takes care of almost all event handling and flow of control, and the programmer's code is called only when the framework needs it (e.g., to create or manipulate a proprietary data structure).

A programmer writing a framework program not only relinquishes control to the user (as is also true for event loop programs), but also relinquishes the detailed flow of control within the program to the framework. This approach allows the creation of more complex systems that work together in interesting ways, as opposed to isolated programs, having custom code, being created over and over again for similar problems.

Thus, as is explained above, a framework basically is a collection of cooperating classes of objects that make up a reusable design solution for a given problem domain. It typically includes objects that provide default behavior (e.g., for menus and windows), and programmers use it by inheriting some of that default behavior and overriding other behavior so that the framework calls application code at the appropriate times. There are three main differences between frameworks and class libraries:

Behavior versus Protocol. Class libraries are essentially collections of behaviors that you can call when you want those individual behaviors in your program. A framework, on the other hand, provides not only behavior but also the protocol or set of rules that govern the ways in which behaviors can be combined, including rules for what a programmer is supposed to provide versus what the framework provides.

Call versus Override. With a class library, the programmer codes objects and calls their member functions. It's possible to code and call objects in the same way with a framework (i.e., s to treat the framework as a class library), but to take full advantage of a framework's reusable design, a programmer typically writes code that overrides and is called by the framework. The framework manages the flow of control among its objects. Writing a program involves dividing responsibilities among the various pieces of software that are called by the framework rather than specifying how the different pieces should work together.

Implementation versus Design. With class libraries, programmers reuse only implementations, whereas with frameworks, they reuse design. A framework embodies the way a family of related programs or pieces of software work. It represents a generic design solution that can be adapted to a variety of specific problems in a given domain. For example, a single framework can embody the way a user interface works, even though two different user interfaces created with the same framework might solve quite different interface problems.

Thus, through the development of frameworks for solutions to various problems and programming tasks, significant reductions in the design and development effort for software can be achieved.

To date, Web development tools have been limited in their ability to create dynamic Web applications that span from client to server and inter-operate with existing computing resources. Until recently, HTML has been the dominant technology used in development of Web-based solutions. However, HTML has proven to be inadequate in the following areas: poor performance; restricted user interface capabilities; lack of interoperability with existing applications and data; inability to scale, and weak security.

Sun Microsystem's Java language solves many problems by: improving performance; enabling the creation of dynamic, real-time web applications; and providing the ability to create a wide variety of user interface components.

With Java, developers can create robust User Interface (UI) components. Custom "widgets" (e.g. real-time stock tickers, animated icons, etc.) can be created, and performance is improved. Unlike HTML, Java supports the notion of validation, offloading appropriate processing onto the client for improved performance. Dynamic, real-time Web pages can be created. Using the above-mentioned custom UI components, dynamic Web pages can also be created.

Sun's Java language has emerged as an industry-recognized language for "programming the Internet." Sun defines Java as: "A simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded, dynamic, buzzword-compliant, general-purpose programming language. Java supports programming for the Internet in the form of platform-independent Java applets." Java applets are small, specialized applications that comply with Sun's Java Application Programming Interface (API) allowing developers to add "interactive content" to Web documents (e.g. simple animations, page adornments, basic games, etc.). Applets execute within a Java-compatible browser (e.g. Netscape Navigator or Internet Explorer) by copying code from the server to client. From a language standpoint, Java's core feature set is based on C++. Sun's Java literature states that Java is basically "C++, with extensions from Objective C for more dynamic method resolution".

Another technology that provides similar function to JAVA is provided by Microsoft and ActiveX Technologies, to give developers and Web designers the wherewithal to build dynamic content for the Internet and personal computers. ActiveX includes tools for developing animation, 3-D virtual reality, video and other multimedia content. The tools use Internet standards, work on multiple platforms, and are being supported by over 100 companies. The group's building blocks are called ActiveX Controls, small, fast components that enable developers to embed parts of software in hypertext markup language (HTML) pages. ActiveX Controls work with a variety of programming languages including Microsoft Visual C++, Borland Delphi, Microsoft Visual Basic programming system and Microsoft's development tool **10** for Java, code named "Jakarta." ActiveX Technologies also includes ActiveX Server Framework, allowing developers to create server applications. One of ordinary skill in the art readily recognizes that ActiveX could be substituted for JAVA without undue experimentation to practice the invention.

SUMMARY OF THE INVENTION

According to a broad aspect of a preferred embodiment of the invention, secure transmission of data is provided between at least two computer systems over a public communication system, such as the Internet. Secure transmission of data is provided from the payor computer system to a banking computer system, which may initiate further secure transmission of payment information regarding a payment instrument from the banking computer system to a the payee computer system. The payment system formats transaction information appropriately and transmits the transaction to the particular host system. The host system evaluates the payment information and returns a level of authorization of credit transfer to the payee computer.

The novel features that are considered characteristic of the invention are set forth with particularity in the appended claims. The invention itself, however, both as to its structure and its operation together with the additional object and advantages thereof will best be understood from the following description of the preferred embodiment of the present invention when read in conjunction with the accompanying drawings. Unless specifically noted, it is intended that the words and phrases in the specification and claims be given the ordinary and accustomed meaning to those of ordinary skill in the applicable art or arts. If any other meaning is intended, the specification will specifically state that a special meaning is being applied to a word or phrase. Likewise, the use of the words "function" or "means" in the Description of Preferred Embodiments is not intended to indicate a desire to invoke the special provision of 35 U.S.C. §112, paragraph 6 to define the invention. To the contrary, if the provisions of 35 U.S.C. §112, paragraph 6, are sought to be invoked to define the invention(s), the claims will specifically state the phrases "means for" or "step for" and a function, without also reciting in such phrases any structure, material, or act in support of the function. Even when the claims recite a "means for" or "step for" performing a function, if they also recite any structure, material or acts in support of that means of step, then the intention is not to invoke the provisions of 35 U.S.C. §112, paragraph 6. Moreover, even if the provisions of 35 U.S.C. §112, paragraph 6, are invoked to define the inventions, it is intended that the inventions not be limited only to the specific structure, material or acts that are described in the preferred embodiments, but in addition, include any and all structures, materials or acts that perform the claimed function, along with any and all known or later-developed equivalent structures, materials or acts for performing the claimed function.

DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages are better understood from the following detailed description of a preferred embodiment of the invention with reference to the drawings, in which:

FIG. **1** is a block diagram of a representative hardware environment in accordance with a preferred embodiment;

FIG. **2** depicts an overview in accordance with a preferred embodiment;

FIG. **3** is a block diagram of the system in accordance with a preferred embodiment;

FIG. **4** depicts a preferred embodiment of an acceptance method according to the present invention.

DETAILED DESCRIPTION

The present invention is a system or method and device useful for the secure electronic payment of consumer debts over a publicly accessible computer network.

A preferred embodiment of a system in accordance with the present invention is practiced in the context of personal computers or workstations. A representative hardware environment is depicted in FIG. **1**, which illustrates a typical hardware configuration of a computer workstation in accordance with a preferred embodiment having a central processing unit **10**, such as a microprocessor, and a number of other units interconnected via a system bus **12**. The workstation shown in FIG. **1** includes Random Access Memory (RAM) **14**, Read Only Memory (ROM) **16**, an I/O adapter **18** for connecting peripheral devices, such as disk storage units **20** to the bus **12**, a user interface adapter **22** for connecting a keyboard **24**, a mouse **26**, a speaker **28**, a

microphone **32**, and/or other user interface devices, such as a touch screen or and the like (not shown) to the bus **12**, communication adapter **34** for connecting the workstation to a communication network **100** (e.g., a data processing network) and a display adapter **36** for connecting the bus **12** to a display device **38**. The workstation typically has resident thereon an operating system such as the Microsoft Windows NT or Windows/95 Operating System (OS), the IBM OS/2 operating system, the MAC OS, or UNIX operating system. Those skilled in the art will appreciate that the present invention may also be implemented on platforms and operating systems other than those mentioned.

The preferred embodiment of the invention utilizes a variety of different software languages, (preferably CC++ and JAVA but may include such languages as HyperText Markup Language (HTML) and Extended Markup Language (XML)), to implement objects and documents on the Internet **100** together with a general-purpose secure communication protocol for a transport medium between the client and the payee.

FIG. **3** depicts an overview of the present invention. The preferred form of the present invention uses at least two separate, but compatible, software packages. Security server software that continuously runs on a security server **140** and payor software that runs on demand on a payor computer system **120**. The payor computer system **120** communicates via the payor software with the security server **140** via the security server software. The communication, or transaction, session operates under the secure communication protocol described below. A payee computer system **130** may also communicate via payee software with the security server **140**.

A security server **140** is a computer system that provides electronic commerce services in support of a bank or other financial institution, and that interfaces to the financial institution to support the authorization and capture of transactions. The transaction session between the payor computer system **120** and the security server **140** operates under a variant of a secure payment technology, as described herein, referred to as Payor-Originated Secure Electronic Transactions ("POSET"), as is more fully described herein.

Initially, the payor creates a one-time payor profile at the receiving institution, such as a bank. The payor profile preferably includes a user pass phrase and user created personal verification questions, which are used for future verification of payor identity. The verification questions are randomly created by the payor and may be questions such as mother's maiden name, favorite color or the like. Preferably the pass phrase is not limited to a short contiguous number and/or letter combination, like an ordinary pin or password, but can include blank or white spaces between characters or words. The use of a longer phrase helps to prevent a "dictionary attack" on the pass phrase. The benefit to the use of a phrase with white spaces is the increased ease with which the payor can remember a more complicated pass-phrase, thereby increasing security. The payor profile information is encrypted and made resident on the security server **140**. The encryption of the payor profile information adds a level of security against unauthorized access by institution or bank personnel.

Generally, a security server program is resident and continuously running on the security server **140**. This allows the security server program to be accessed at any time by a payor. The payor launches the payor software program on the payor computer system **120**. Upon the first launch of the payor program, the payor program executes an initialization, or registration procedure. In this procedure, the payor computer program requires the payor to create and enter a customer, or access, password. This password is used on each subsequent launch of the payor program to verify and identify the payor and account or accounts being accessed. A user transaction screen, such as a "check screen" is then presented to the payor. In the transaction screen are transaction fields that the payor fills in and sends to the security server program. Once a transaction has been initialized, the payor computer program then attempts to contact the security server **140** over a computer network system, commonly known as the Internet **100**.

During the first session, after communications have been established between the security server **140** and the payor computer system **120**, in one embodiment the security server program obtains hardware specific information from the payor computer system **120**. Since hardware specific information is individual to each computer, it acts as a "finger print" that can be used to uniquely identify the computer program. Additionally, this allows easy verification and identification of the specific computer program during future transactions. The hardware specific information is encrypted and stored with the payor profile information in the security server database or on both computers to account for subsequent changes in the hardware configuration, i.e., new hard drives and the like. By combining use of the customer password, the payor computer hardware information, and payor profile information, both the payor and the payor's program may be quickly, easily, and securely verified during future transactions.

When initiating communication with the security server system **140**, the payor computer system **120** may use any well-known access protocol, e.g., Transmission Control Protocol/Internet Protocol ("TCP/IP"). A description of TCP/IP is provided in Information Sciences Institute, "Transmission Control Protocol DARPA Internet Program Protocol Specification (RFC 793)" (September 1981), and Information Sciences Institute, "Internet Protocol DARPA Internet Program Protocol Specification (RFC 791)" (September 1981). In this implementation, the payor computer system **120** acts as a client and the security server system **140** acts as a server.

When initiating communication with the security server **140**, the payor computer system **120** first sends a "client request for connection" message to the security server system **140**. The client request for connection message may further include a variable length session identifier.

In response to the client request for connection message, if the security server system **140** wishes to correspond with the payor computer system **120**, it responds with a message to the payor computer system **120** to switch to a second, separate transaction port, thereby creating a second, or transaction, session. An alternate way to consider this is as a single session that is conducted over two separate pathways: 1) over a first initial communications port; and 2) over a second transaction port. This is an important part of the present invention in that the identity of the second port is dynamically assigned and changes for each and every independently created session. This creates an extra element of variability to the transaction process that improves to the security of the transaction. If the security server system **140** does not wish to communicate with the payor computer system **120**, it responds with a message indicating refusal to communicate.

FIG. **4** depicts the detailed steps of authorizing communications between the payor computer **120** and the security

server **140**, including the generating and transmitting of a payment authorization request.

Preferably, the security server connection message includes an initial very large prime number, a prime modulus, and a server calculated intermediate value. The payor computer system **120** replies to the security server connection message with a client response message that preferably includes a payor calculated intermediate value. Separately, the payor computer system **120** calculates a final shared value. Once the security server **140** receives the payor calculated intermediate value, it too calculates the final shared value.

More specifically, the security server system **140** obtains a randomly generated server secret number. The security server **140** also selects a very large public prime number, which is preferably, a very large prime number residing in a pre-created prime number database, and a prime modulus. The security server **140** creates a server calculated intermediate value using the secret random number, the public prime number, and the prime modulus, by performing a portion of a selected algorithm. The security server **140** sends the server calculated intermediate value, the public prime number, and the prime modulus to the payor computer system **120**. The payor computer system **120** generates a payor secret random number and uses the public prime number and prime modulus to create a payor calculated intermediate value. Additionally, the payor computer system **120** uses the server calculated intermediate value and the payor calculated intermediate value to calculate a shared final value. The payor computer system **120** sends the payor calculated intermediate value, along with selected payor ID or profile information to the security server **140**. In a preferred embodiment, the calculated intermediate values are encrypted before transmission. The security server **140** uses the payor calculated intermediate value, with the security server calculated intermediate value, to also calculate the same shared final value. Thus, the shared final value, which while known by both computers, is never transmitted. For this very reason, the shared final value cannot be intercepted by a third party for use in a fraudulent attempt on the account.

While other like algorithms with similar properties may be used, a preferred algorithm for the above security process is as follows:

SIV (security server intermediate value)=g^{SRN} mod p,

where g is the public prime number, mod p is the prime modulus, and SRN is the security server random number,

CIV (payor intermediate value)=g^{CRN} mod p,

where g is the public prime number, mod p is the prime modulus, and CRN is the payor random number; and

SFV (shared final value)=SIV^{CRN} mod p, for the payor computer **120**, and

=CIV^{SRN} mod p, for the security server **140**.

In creating the payor random number, a preferred embodiment has the payor computer system **120** using random values to seed the creation of a unique large random number. Preferably, these random numbers are obtained dynamically within the session, and even more preferably, are obtained from unique non-repeatable functions, such as mouse or cursor positions.

In each separate system, the payor computer system **120** and the security server **140**, the shared final value, in combination with selected portions of the payor profile information, is encrypted using yet another function, such as a one-way secure hash algorithm, to produce a final session key. This final session key, having portions that are generated dynamically within each session, and portions that are personal to each individual payor, is computationally impossible to decode or generate in the time allotted for each transaction, thereby providing an exceptionally high level of security for each transaction. The inclusion of a one-way function encryption of the data provides an ultra-high level of security for each transaction.

The security server **140** then takes the payor's ID information and compares to the payor profile information resident in the security server **140** in order to verify the payor. If the payor ID information is not verified, the connection is immediately terminated and logged to a failure database.

If the payor profile information is verified during the transaction, the security server **140** proceeds to check several flags, or indicia fields, for an indication of whether further verification or encryption, such as by providing the answers to the private questions previously supplied, is required.

At this point both the payor computer system **120** and the security server system **140** have: 1) negotiated a communication session; 2) have communicated to each other the basis for the calculation of a set of encryption keys that may be used to encrypt and decrypt further communications between the two computer systems, **120** and **140** respectively; and 3) have calculated a final session key that is never transmitted and will be used for further encryption. The payor computer system **120** and the security server system **140** may thereafter engage in a secure financial transaction with a greatly reduced risk of interception or fraud by third parties.

After a connection has been authorized and implemented, the security server **140** checks the present account register number of the payor program. Initially, the current account register number is set to a zero transaction number. At any time, including immediately after registration of the program but before a first transaction has been processed, the payor may simply elect to exit the payor computer program and reenter it at another time. If the payor does not exit the payor program, a graphical user program interface, as discussed above, preferably the "check" screen, is generated on the payor computer **120**. The program interface includes enterable fields for transaction specific information.

If additional encryption is requested, the security server **140** may request portions of transaction information previously sent or it may encrypt a verifying value. If the verifying value is sent to the payor computer system **120**, the payor computer system **120** decrypts the verifying value and uses it in one of two different ways. First, it may be used as additional data added to the transaction information, re-encrypted and sent to the security server. Then, when the security server **140** decrypts the transaction it compares the verifying value before processing the transaction. Second, the decrypted verifying value may be used as an initial value to roll-over encrypt the transaction information, which is then further encrypted using the final session key and sent to the security server **140**. The security server **140** then decrypts the message, and decrypts the roll-over encrypted transaction (using the verifying value). If the both systems use the same verifying value, the security server then has transaction information that is appropriate for the system, otherwise the decryption of the rolled-over information will yield strange characters and/or information. These two methods are typically selected by the software and may be dynamically chosen such that any individual transaction may use one or the other method.

Once the transaction has been verified and processed, the security server **140** creates a time stamp, encrypts it, and

sends it to the payor computer system **120** to finalize the transaction. In this way, the payor, not being in control of the time stamp, cannot create a false time record. Once the payor program receives the time stamp, it then increments the account register number counter of the payor program by one and fills in the check information.

Among the information communicated by the payor computer system **120** to the security server system **140** may be information that specify payment information, such as payee identification, bank identification, bank account numbers, credit card numbers, and related information, collectively referred to as "payment information," that may be used to pay the bill for the goods and/or services ordered. In order to obtain payment, the payee may supply a portion of this information to the bank or other institution responsible for the proffered payment method. This enables the payee to perform payment authorization and payment capture. Payment authorization is the process by which permission is granted to a security server **140** operating on behalf of a financial institution to authorize payment on behalf of the financial institution. This is a process that assesses transaction risk, confirms that a given transaction does not raise the account holder's debt above the account's balance. Payment capture is the process that triggers the movement of funds from the financial institution to the payee's account in order to settle the account.

The security server system **140** identifies the payee for which the transaction is authorized by inspection of the transaction information. The security server system **140** may contact the appropriate payee using a secure means, preferably via the Internet, and using prior art means, obtains a response indicating whether the requested payment is due, presented, and has been confirmed.

In contacting the payee, the security server may utilize one of two different methods. A first method is used for non-institutional payee's, such as private individuals or small businesses. In this method, the security server program automatically generates an electronic mail message (e-mail) that identifies the payor and the fact that a payment has been made. It is preferable that the e-mail message does not indicate the amount of payment or account to which the payment was made for security purposes. A second method, which is preferably used for larger payee's such as large business and institutions is the use of a payee program on a payee computer system **130**. The payee program communicates with the security server program, as detailed below, and may provide, among other information, the name of the payor, the invoice number or customer number, the amount of payment, the account to which the payment has been made, and the like. The transaction between the payee computer program and the security server computer program may be accomplished in either a batch mode or in a continuous, real-time action.

Upon the first launch of the payee program, the payee program executes an initialization, or registration procedure. In this procedure, the payee computer program requires the payee to create and enter a payee, or access, password. This password is used on each subsequent launch of the payee program to verify and identify the payee and account or accounts being accessed. The payee computer system **130** then contacts the security server **140** over a computer network system, commonly known as the Internet **100**. The payee program communicates with the security server program and registers the payee program with the security server program. This registration confirms the identity of the payee computer program.

During a first transaction session, after communications have been established between the security server **140** and the payee computer system **130**, the security server **140** preferably obtains hardware specific information from the payee computer system **130** and stores it in both places to account for changes in the hardware configuration of the payee computer **130**. Since hardware specific information is individual to each computer, it acts as a "finger print" that can be used to uniquely identify the computer. Additionally, this allows easy verification and identification of the specific computer during future transactions. The hardware specific information is encrypted and stored with the payee ID information in the security server database. By combining use of the payee password, the payee computer hardware information, and payee profile information, both the payee and the payee's computer program may be quickly, easily, and securely verified during future transactions.

When initiating communication with the security server system **140**, the payee computer system **130** may use any well-known access protocol, e.g., Transmission Control Protocol/Internet Protocol ("TCP/IP"). A description of TCP/IP is provided in Information Sciences Institute, "Transmission Control Protocol DARPA Internet Program Protocol Specification (RFC 793)" (September 1981), and Information Sciences Institute, "Internet Protocol DARPA Internet Program Protocol Specification (RFC 791)" (September 1981). In this implementation, the payee computer system **130** acts as a client and the security server system **140** acts as a server. It should be noted that the communication may be initiated by the security server program to the payee program with the security server system **140** acting as the client and the payee computer system **130** acting as the server.

When initiating communication with the security server **140**, the payee computer system **130** first sends a "payee request for connection" message to the security server system **140**. The payee request for connection message may further include a variable length session identifier.

In response to the payee request for connection message, if the security server system **140** wishes to correspond with the payee computer system **130**, it responds with a message to the payee computer system **130** to switch to a second, separate transaction port, thereby creating a second, or transaction, session. Another way of thinking about this is as a single session with two separate pathways: 1) a first port for initializing communications; and 2) a second port for transmission of transaction information. This is an important part of the present invention in that the identity of the second port is dynamically assigned and changes for each and every independently created session. This creates an extra element of variability to the transaction process that improves to the security of the transaction. If the security server system **140** does not wish to communicate with the payee computer system **130**, it responds with a message indicating refusal to communicate.

FIG. **4** depicts the detailed steps of authorizing communications between the payee computer **130** and the security server **140**, including the generating and transmitting of a payment authorization request.

Preferably, the security server connection message includes an initial very large prime number, a prime modulus, and a server calculated intermediate value. The payee computer system **130** replies to the security server connection message with a payee response message that preferably includes a payee calculated intermediate value. Separately, the payee computer system **120** calculates a final shared value. Once the security server **140** receives the payee calculated intermediate value, it too calculates the final shared value.

More specifically, the security server system **140** obtains a randomly generated server secret number. The security server **140** also selects a public prime number, which is preferably, a very large prime number residing in a pre-created prime number database, and a prime modulus. The security server **140** creates a server calculated intermediate value using the secret random number, the public prime number, and the prime modulus, by performing a portion of a selected algorithm. The security server **140** sends the server calculated intermediate value, the public prime number, and the prime modulus to the payee computer system **130**. The payee computer system **130** generates a payee secret random number and uses the public prime number and prime modulus to create a payee calculated intermediate value. Additionally, the payee computer system **130** uses the server calculated intermediate value and the payee calculated intermediate value to calculate a shared final value. The payee computer system **130** sends the payee calculated intermediate value, along with selected payee ID information to the security server **140**. In a preferred embodiment, the calculated intermediate values are encrypted before transmission. The security server **140** uses the payee calculated intermediate value, with the security server calculated intermediate value, to also calculate the same shared final value. Thus, the shared final value, which while known by both computers, is never transmitted. For this very reason, the shared final value cannot be intercepted by a third party for use in a fraudulent attempt on the account.

While other like algorithms with similar properties may be used, a preferred algorithm for the above security process is as follows:

SIV (security server intermediate value)=g^{SRN} mod p,

where g is the public prime number, mod p is the prime modulus, and SRN is the security server random number;

MIV (payee intermediate value)=g^{MRN} mod p,

where g is the public prime number, mod p is the prime modulus, and MRN is the payee random number; and

SFV (shared final value)=SIV^{MRN} mod p, for the payee computer **130**, and

=MIV^{SRN} mod p, for the security server **140**.

In creating the payee random number, a preferred embodiment has the payee computer system **130** using random values to seed the creation of a unique large random number. Preferably, these random numbers are obtained dynamically within the session, and even more preferably, are obtained from unique non-repeatable functions, such as mouse or cursor positions, line voltages, or the like.

In each separate system, the payee computer system **130** and the security server **140**, the shared final value, in combination with selected portions of the payee profile information, is encrypted using yet another function, such as a one-way secure hash algorithm, to produce a final session key. This final session key, having portions that are generated dynamically within each session, and portions that are personal to each individual payee , is computationally impossible to decode or generate in the time allotted for each transaction, thereby providing an exceptionally high level of security for each transaction. The inclusion of a one-way function encryption of the data provides an ultra-high level of security for each transaction.

The security server **140** then takes the payee's profile information and compares to the payee profile information resident in the security server **140** in order to verify the payee. If the payee profile information is not verified, the connection is immediately terminated and logged to a failure database.

If the payee profile-information is verified during the transaction, the security server **140** proceeds to check several flags, or indicia fields, for an indication of whether further verification or encryption is required.

At this point both the payee computer system **130** and the security server system **140** have: 1) negotiated a communication session; 2) have communicated to each other the basis for the calculation of a set of encryption keys that may be used to encrypt and decrypt further communications between the two computer systems, **130** and **140** respectively; and 3) have calculated a final session key that is never transmitted and will be used for further encryption. The payee computer system **130** and the security server system **140** may thereafter engage in a secure financial transaction with a greatly reduced risk of interception or fraud by third parties.

If additional encryption is requested, the security server **140** uses the final session key and encrypts a verifying value. The verifying value is sent to the payee computer system **130**. The payee computer system **130** decrypts the verifying value. The decrypted verifying value is then used in one of two different ways. First, it may be used as additional data added to the transaction information, re-encrypted and sent to the security server. Then, when the security server **140** decrypts the transaction it compares the verifying value before processing the transaction. Second, the decrypted verifying value may be used as an initial value to roll-over encrypt the transaction information, which is then further encrypted using the final session key and sent to the security server **140**. The security server **140** then decrypts the message, and decrypts the roll-over encrypted transaction (using the verifying value). If the both systems use the same verifying value, the security server then has transaction information that is appropriate for the system, otherwise the decryption of the rolled-over information will yield strange characters and/or information. These two methods are typically selected by the software and may be dynamically chosen such that any individual transaction may use one or the other method.

Once the transaction has been verified and processed, the security server **140** creates a time stamp and sends it to the payee computer system **130** to finalize the transaction. In this way, the payee, not being in control of the time stamp, cannot create a false time record.

For the above payee-security server transaction, the payee computer system **130** generates a payee payment capture request and transmits it to the security server system **140**. The security server **140** processes the payment capture request, generates a payment capture response and transmits it to the payee computer system **130**. The payee computer system **130** processes payment capture response and verifies that payment for the goods or services purchased by the payor have been captured. The basic capture request is a data area that includes all the information needed by the security server system **140** to trigger a transfer of funds to the payee operating the payee computer system **130**.

Specifically, a capture request includes, as a minimum amount of information, a capture request amount, a date, and a Payee ID (MID) for the particular payee.

The security server system **140** creates a basic capture response. The basic capture response is a data area that includes all the information to indicate whether a capture request was granted or denied.

A Virtual Point of Payment (vPOP) software is also described in accordance with a preferred embodiment using smart card technology or kiosk technology. The vPOP software provides payment functionality on independent

platforms, allowing a payor to process payments securely using a smart card and the Internet **100**. It provides full payment functionality for a variety of payment instruments.

A brief description of the vPOP software functions are provided below. The vPOP provides an interface for transactions that are initiated by the consumer. The consumer initiates a transaction from a Graphical User Interface (GUI) and all the transactions that are initiated by the consumer using a smart card and are routed through a remote computer or kiosk to the security server.

The payment functionality provided by the vPOP software is "Consumer-Initiated" at a site remote from the payee computer system **130**. The normal flow of a transaction is via the vPOP software into a security server software that is responsible for converting into the appropriate format for additional processing and forwarding to existing host payment authorization systems.

Smart cards, according to the present invention has a cyclic registry that is used for transaction data storage. There are at least two separate registers in which at least two separate transactions may be stored. The actual number of registers is only limited by the available space in the memory of the smart card hardware. Additionally, each smart card must be registered to each individual at the financial institution, like a credit card, to prevent unauthorized access. This includes the use of a pin number or pass phrase to access the functionality of the smart card. Finally, the smart card may have encrypted verification information, such as portions of the above described payee profile information, which is used by the security server to securely identify the payor.

In use, the smart card is inserted into a kiosk computer having a modified version of the payee computer program running (the modification being the lack of a registry memory function). The payor is required to provide the smart card pin number or pass phrase. Once the payor correctly provides the smart card pin number or pass phrase, the transaction (check) screen is presented with transaction header information, which is encrypted and stored on the smart card, already filled in. The payor fills in the applicable fields and sends the transaction to the security server program using the same encryption and verification process as described above.

If the payor repeatedly provides an incorrect pin number or pass phrase, or if the security server program has the smart card flagged as missing or stolen, then the security server program sends a message to the kiosk computer to keep the smart card and not release it to the user. Alternately, the security server may send a message to the kiosk computer deactivating the smart card at the kiosk computer. In yet another embodiment, the security server periodically uploads to the kiosk computer a list of missing or stolen smart cards. In this embodiment, the kiosk computer reads the identification of the smart card upon insertion and, upon identification of the smart card as flagged, refuses to allow access to the kiosk program or transaction processing and may or may not keep the smart card. Additionally, the security server may communicate the time and location of the use of the stolen smart card to the proper authorities.

Host Payment Functionality: these transactions require communication with the security server **140**, either immediately or at a later stage. For example, an Online Authorization-Only transaction, when initiated, communicates with the host immediately. However, an Off-line Authorization-Only transaction is locally authorized by the vPOP software without having to communicate with the host, but at a later stage this off-line authorization transaction is sent to the host. Within the Host Payment Functionality some transactions have an associated Payment Instrument, while others do not. These two kinds of transactions are:

Host Financial Payment Functionality: these transactions have a Payment Instrument (Smart card, Credit Card, Debit Card, E-Cash, E-Check, etc.) associated with them.

Host Administrative Payment Functionality: these transactions do not require a payment instrument, and provide either administrative or inquiry functionality. Examples of these transactions are "Reconcile" or the "Batch Close."

Local Functions and Transactions: these transactions do not require communication with the host at any stage, and provide essential vPOP software administrative functionality. An example of this is the vPOP software configuration function, which is required to set up the vPOP software. Another example is the "vPOP Batch Review" function, which is required to review the different transactions in the vPOP Batch or the Transaction Log.

A preferred embodiment of the vPOP software supports various Payment Instruments. A consumer chooses a payment based on personal preferences. Some of the Payment Instruments supported include credit cards, debit cards, electronic cash, electronic checks, and micro-payments (electronic coin).

As discussed above, the different Payment Functionality provided by the vPOP terminal require communication with the security server **140** and these transactions are referred to as "Online Transactions." The transactions that can be done locally without having to communicate are referred to as "Local Functions/Transactions." In order to provide support for many different types of Payment Instruments, the vPOP Payment Functionality have been categorized.

An authorization without capture transaction is used to validate the card holder's account number for a payment that needs to be performed at a later stage. The transaction does not confirm a payment's completion to the host, and there is no host data capture in this event. The vPOP captures this transaction record and later forwards it to the host to confirm the payment in a forced post transaction request.

A forced post transaction confirms to a host computer that a completion of a payment has been accomplished and requests data capture of the transaction. The forced post transaction is used as a follow-up transaction after doing an authorization (Online or Off-line) transaction.

The offline post transaction is identical to the "authorization without capture" transaction, except that the transaction is locally captured by the vPOP without initiating communication with a host. A forced post operation is done as a follow-up operation of this transaction.

The Internet **100** provides the communication processing necessary to enable the vPOP software. The software interface CGI communicates via the Internet **100** to provide information to the vPOP Security Server **140**, which formats information in accordance with the vPOP.

As discussed above, in order to actually transact business over the Internet **100**, the user must first register the smart card with the bank with which he signs an acquiring agreement. For online payment processing, the user must also create an appropriate set of digital credentials in the form of personal questions and possibly additional passwords, depending on the financial institution and/or user's desires.

The user, interacting with the software, can navigate to a list of security servers, and selects the bank to connect to by selecting from the list of banks.

Each vPOP may process a single transaction at a time. Security Servers **140** can process many transactions at a

time, so transaction requests can often occur simultaneously at the security server **140**. Thus, the security server **140** version of the vPOP Software must have support for multi-tasking and provide support for multiple threads to be active at the same time in the same system as well as the same process. This requirement is relatively straightforward.

Since the Internet **100** is so pervasive, and access is available from virtually any computer, utilizing the Internet **100** as the communication backbone for connecting the payor, payee and access to the authorizing bank through a security server **140** allows the payee vPOP software to be remotely located from the payee's premises. For example, the payor could pay for goods from any computer system attached to the Internet **100** at any location in the world. Similarly, the payee vPOP system could be located at a central host site where payee vPOP systems for various payees all resided on a single host with their separate access points to the Internet **100**. The payee could utilize any other computer attached to the Internet **100** utilizing a protocol to query the remote vPOP system and obtain capture information, payment administration information, inventory control information, audit information and process payor satisfaction information. Thus, without having to incur the overhead of maintaining sufficient computer processing power to support the vPOP software, a payee can obtain the information necessary to run a business smoothly and avoid hiring IS personnel to maintain the vPOP system.

A novel feature of the vPOP software provides payment page customization based on a user's preferences. This feature automatically lists cards that are held by the user and accepted by particular payees based on the active terminal configuration. Each approved card for a particular payee provides smart card information supported by the payee.

Because the security server **140** must sustain reliable operations and enable graceful evolution, it is designed with some important attributes, including: security, availability, performance, scalability, and manageability.

Site redundancy and location redundancy allows the security server **140** to sustain service through virtually instantaneous recovery from internal failures or external disasters that cause physical damage to the system. Minimum-outage recovery is possible with redundant configurations of important components.

The security server **140** supports connections to a proprietary bank network and supports mirrored disk arrays.

The security server **140** architecture supports location redundancy where a secondary remote system is connected to the primary system via dedicated WAN links for software-driven database duplication.

The security server **140** software architecture, the choice of third-party software components, and the selection of hardware platforms enable the security server **140** to gracefully adapt and evolve to take on new demands in different dimensions.

The encryption and decryption algorithms used in processing the messages require significant computational power. A "security processor" is deployed with the security server **140** to boost the performance of cryptographic algorithms. The security processor is a networked peripheral device to the security server **140**. It provides cryptographic services suitable for processing, and its services are accessible via calls to software libraries.

Security server **140** statistics about transaction requests (by transaction type) and transaction results (e.g., success, failed due to host, failed due to authentication, etc.) can be determined at any time for a particular time interval by generating a report.

A replay attack at the security server **140** is a request where either: a) the request is stale; the request was received "too late" with respect to the reqdate in the request (this window is specified by a configurable security server policy); b) the request is not stale but the exact Request/Response pair Id has already been seen before in a request and still logged in the security server **140** database.

If the vPOP times-out for any reason, it must retry later using a Request/Response Pair Id that indicates a new attempt. If the Gateway receives a late-response (after vPOP has given up) it simply logs it in the database for that retry attempt (if no retry attempt for the transaction is still outstanding at the host). There is a glare situation where the original response could arrive so late that it could be confused with the response from a currently outstanding retry attempt with the host. This situation is logged and the first response not sent back to vPOP.

Finally, the method and apparatus described above may be adapted to process transactions for medical records, prescriptions, audio-visual files, court documents, and any other sensitive or confidential information.

The preferred embodiment of the invention is described above in the Drawings and Description of Preferred Embodiments. While these descriptions directly describe the above embodiments, it is understood that those skilled in the art may conceive modifications and/or variations to the specific embodiments shown and described herein. Any such modifications or variations that fall within the purview of this description are intended to be included therein as well. Unless specifically noted, it is the intention of the inventor that the words and phrases in the specification and claims be given the ordinary and accustomed meanings to those of ordinary skill in the applicable art(s). The foregoing description of a preferred embodiment and best mode of the invention known to the applicant at the time of filing the application has been presented and is intended for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and many modifications and variations are possible in the light of the above teachings. The embodiment was chosen and described in order to best explain the principles of the invention and its practical application and to enable others skilled in the art to best utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A method for a secure transaction over a multi-computer network comprising the steps of:

a. providing at least two separate computer programs that are designed to communicate with each other over a multi-computer network, each separate computer program resident and runnable on a separate computer of the multi-computer network, at least one of the at least two separate computer programs further being a security server program for receiving and processing the secure transaction and at least one of the at least two separate computer programs further being a customer program;

b. running the security server program on a substantially continuous basis thereby making it available to receive secure transactions;

c. running the customer program on an as needed basis for communicating with the security server program with the customer program across a first communication port;

d. receiving a dynamically assigned port address from the security server program, further, receiving from the

security server program a public set of numbers and a security server intermediate value that was calculated using at least the public set of numbers;

e. switching the customer program to the dynamically assigned port address for further communications with the security server program;

f. having the customer program calculate a customer intermediate value using at least the public set of numbers and a shared final value using at least the customer intermediate value and the security server intermediate value;

g. sending the customer intermediate value to the security server program;

h. having the security server program calculate the shared final value using the customer intermediate value and the security server intermediate value;

i. having both the security server program and the customer program create an encryption key using at least the shared final value;

j. having the customer computer encrypt transaction information using the encryption key;

k. sending the encrypted transaction information to the security server program;

l. having the security server program de-crypt the encrypted transaction information; and

m. having the security server program process the transaction.

2. The method according to claim **1** wherein the public set of numbers is at least a public prime number and a prime modulus number.

3. The method according to claim **2** wherein the customer intermediate value is further calculated using a customer selected random number and the security server intermediate value is calculated using a security server selected random number.

4. The method according to claim **3** wherein the shared final value is calculated by the customer computer program using at least the security server intermediate value, the customer selected random number, and the prime modulus; and the shared final value is calculated by the security server program using at least the customer intermediate value, the security server selected random number, and the prime modulus.

5. The method according to claim **4** wherein the step of creating an encryption key using at least the shared final value comprises at least the step of passing at least a portion of the shared final value through a further encryption algorithm.

6. The method according to claim **5** wherein the further encryption algorithm is a one-way function.

7. The method according the claim **6** further including the step of having the customer computer program send customer profile information to the security server program for comparison with customer profile information previously stored on a computer memory accessibly by the security server program, thereby verifying the identity of the customer.

8. The method according the claim **1** further including the step of having the customer computer program send customer profile information to the security server program for comparison with customer profile information previously stored on a computer memory accessibly by the security server program, thereby verifying the identity of the customer.

9. The method according to claim **7** wherein the customer profile information comprises a pass phrase that may have white spaces and answers to customer created personal information questions.

10. The method according to claim **8** wherein the customer profile information comprises a pass phrase that may have white spaces and answers to customer created personal information questions.

* * * * *

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Inglis	Art Group:	2768
Serial No.:	09/540,193	Examiner:	Weisberger, Richard D.
Filed:	March 30, 2000		
For:	System, method, and article of manufacture for secure transactions utilizing a computer network		
Atty. Docket No.:	515-001		

Assistant Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Change of Correspondence Address

Dear Sir:

Please change the correspondence address in the above matter to:

Kristofer E. Halvorson
The Halvorson Law Firm, P.C.
1757 E. Baseline Rd. Ste 130
Gilbert, Arizona 85233
(480) 892-2037

Respectfully submitted,

Date: 11/09/04

Kristofer Halvorson, Reg. No. 39,211
The Halvorson Law Firm, P.C.
Attorneys for Applicant
1757 E. Baseline Rd. Ste 130
Gilbert, Arizona 85233
(480) 892-2037

AGREEMENT

This Agreement (the "**Agreement**") is made on _____, 2005, by and among **VPI Acquisition, Inc.**, a company established under the laws of the State of Delaware, with offices located at 304 East Fairview Street, Suite 203, Arlington Heights, IL 60005 (the "**Purchaser**"), **Intelagents, Inc.**, a company established under the laws of the State of Delaware, with offices located at 304 East Fairview Street, Suite 203, Arlington Heights, IL 60005 ("**Intelagents**"), **Verify and Protect, Inc.**, a company established under the laws of the State of Delaware, with offices located at 3333 Warrenville Road, Suite 200 Lisle, IL 60532 ("**VPI**"), **VPI Target, Inc.**, a company established under the laws of the State of Delaware, with offices located at 3333 Warrenville Road, Suite 200 Lisle, IL 60532 ("**Seller**") and all the shareholders of VPI (the "**VPI Shareholders**"). The principal contact for VPI and Seller is Jeffry Boston (the "**Seller Principal**"), or any other person identified to all the parties in writing. Purchaser, Intelagents, VPI, Seller and the VPI Shareholders constitute the **"Parties**" to this Agreement.

RECITALS:

Whereas, VPI is in the business of developing, designing and producing user authentication and data encryption software;

Whereas, VPI designed and produced and patented DeepCloak™, a self-contained technology that authenticates users, generates all keys dynamically, encrypts data and creates digital signatures without use of digital certificates or third party certificate authorities (collectively the "**Products**");

Whereas, Intelagents currently maintains a website at "www.intelagentsinc.com;" and VPI currently maintains a website at "www.verifyandprotect.com;"

Whereas, the Products are designed and produced to provide software developers, device makers and corporation IT departments the best means to secure their applications with the DeepCloak™ technology;

Whereas, Purchaser is a wholly owned subsidiary of Intelagents;

Whereas, Intelagents is in the business of providing advanced wired and wireless products and services for protection of fixed and mobile assets against theft, tampering or terrorist attacks using nuclear, chemical or biological materials;

Whereas, immediately prior to the closing of the transactions contemplated by this Agreement, VPI shall contribute to Seller all of the assets of VPI and Seller shall assume certain liabilities of VPI;

Whereas, this Agreement contemplates a tax-free merger of Seller into Purchaser in a reorganization pursuant to Section 368(a)(1)(A) of the Internal Revenue Code.

Now, Therefore, for and in consideration of the mutual covenants contained herein, it is agreed by and between the parties as follows:

1. General Provisions

1.1 The foregoing recitals are incorporated as an integral part of this Agreement.

1.2 The VPI Shareholders agree: (a) within ___ days following the date of this Agreement, to provide to the Purchaser a complete and fully executed Personal and Business Representation for the VPI Shareholders which is attached hereto as **Schedule 1** to this Agreement and which the VPI Shareholders agree to update with material changes thereto as they become known to the VPI Shareholders; (b) to provide the Purchaser and its officers with status reports regarding VPI's ongoing business activities; (c) not to divulge or appropriate for their own use, or for the use of others any Confidential Information (as defined in Section 1.3 below) without the Purchaser's prior written authorization; and (d) that all the Purchaser's and Intelagents' software, hardware, documents, work papers and products shall remain the sole and exclusive property of the Purchaser and Intelagents, and will be returned to the Purchaser or Intelagents in the event the transactions contemplated by this agreement are not consummated or are terminated pursuant to this Agreement.

1.3 Each Party acknowledges that by virtue of this Agreement, it will be exposed to certain confidential information belonging to the other Parties, including but not limited to information about their respective general business operations, internet technology, software, hardware, codes, client information, budgets, debts, tax identification numbers, and other proprietary information (collectively referred to as "**Confidential Information**"). Confidential Information does not include information that (a) was previously known by a Party at the time of receipt, (b) was in the public domain at the time of receipt or thereafter entered the public domain without fault of the recipient, (c) corresponds to information which was furnished to the recipient by a third party lawfully entitled to do so, (d) was developed independently by personnel of the recipient who had no access to the information or (e) is required to be disclosed in legal proceedings. The Parties agree not to use, copy, remove from the control of the Party owning the Confidential Information, or divulge any Confidential Information at any time except as required or permitted by this Agreement. The Parties further agree to take all reasonable steps necessary, or reasonably requested by the other Parties, to ensure that all Confidential Information is kept confidential. The Parties also agree that the restrictions set forth herein are necessary and reasonable to protect each other's legitimate and protectable interests. The Parties agree to be responsible for any breach of this provision by any of their respective representatives regardless of whether such breach occurs in or outside the course of their services on behalf of said Party. In the event there is a termination of this Agreement, all Confidential Information shall be immediately returned to the Party owner.

1.4 The Parties further agree that they (a) shall not, directly or indirectly, dispute or contest the strictly confidential nature of the Confidential Information as disclosed to them or as developed by them in their work with another Party; (b) shall not, directly or indirectly, assist any third party in disputing or contesting the same; and (c) shall not, directly or indirectly,

circumvent, or ask or cause or allow any third party to circumvent the Confidential Information of any Party hereto.

1.5 The Parties acknowledge that any unauthorized disclosure of the Confidential Information of any Party hereto, written or verbal, could cause substantial and irreparable harm to such Party, jeopardize the security programs of its customers and endanger the safety of the public. Therefore, the Parties agree that in the event they breach this covenant of confidentiality, non-disclosure, non-circumvention and non-compete or cause or allow any third parties to do so, the damaged Party shall have, in addition to other remedies available in event of a breach of this covenant, the right to injunctive, punitive or other equitable relief; and, that the prevailing party shall be entitled to recover costs and reasonable attorneys fees. This provision pertains to all Confidential Information, existing documents and those created in the future.

2. The Merger

2.1 The Merger. Subject to the terms and conditions hereof, at the Effective Time (as defined in Section 2.2 below), Seller shall be merged with and into the Purchaser (the **"Merger"**) in accordance with the applicable provisions of the General Corporation Law of the State of Delaware ("**DGCL**"), and the Purchaser, as the surviving corporation in the Merger (in such capacity, the **"Surviving Corporation"**), shall continue its corporate existence, and the separate existence of Seller shall thereupon cease and its corporate existence shall be merged into and transferred to the Surviving Corporation.

2.2 Consummation of the Merger; Effective Time. On the Closing Date (as herein defined) and subject to the satisfaction and performance of the terms and conditions of this Agreement, the Parties will file with the Secretary of State of Delaware a certificate of merger and other documents (the "**Merger Documents**"), in such respective forms as required by, and executed in accordance with, the relevant provisions of the DGCL in order to effect the Merger. The Merger shall become effective at such time as the Merger Documents shall have been accepted for filing with the Secretary of State of Delaware or such other times and dates as the Parties shall agree should be specified in the Merger Documents (the "**Effective Time**").

2.3 Effect of the Merger.

(a) At the Effective Time, the Surviving Corporation shall succeed, without any other action, to all rights and property of Seller and shall be subject to all the debts and liabilities of Seller in the same manner as if the Surviving Corporation had itself incurred them, all with the effect set forth in the DGCL.

(b) At the Effective Time, the certificate of incorporation and the bylaws of the Purchaser, as in effect immediately prior to the Effective Time, shall thereafter continue in full force and effect as the certificate of incorporation and the bylaws of the Surviving Corporation until thereafter changed or amended as provided therein and by the DGCL.

(c) At the Effective Time, the officers and directors of the Purchaser, as in effect immediately prior to the Effective Time, shall thereafter continue as the officers and

directors of the Surviving Corporation to serve in accordance with the bylaws of the Surviving Corporation.

2.4 Merger Consideration. The total consideration (the "**Merger Consideration**") to be paid in connection with the transactions contemplated by this Agreement shall be as set forth below:

(a) An amount equal to Two Million Two Hundred Fifty Thousand Dollars ($2,250,000) (the "**Cash Merger Consideration**"), evidenced by a Promissory Note in similar form and substance as attached hereto as **Exhibit A** (the "**Promissory Note**"), to be paid in cash in accordance with the following schedule:

(i) For each incremental Five Million Dollars ($5,000,000) of equity funding and/or customer contract payments received by the Purchaser, the VPI Shareholders shall receive payment of $500,000; provided, that, that the first $500,000 payment will be paid upon the receipt by the Purchaser of $4,300,000 in funding and/or customer contract payment.

(ii) The payments set forth in (i) above shall continue until such time as an aggregate amount of Two Million Two Hundred Fifty Thousand Dollars ($2,250,000) has been paid.

(b) Three Million Fifty Thousand (3,050,000) shares of the capital stock of Intelagents (the **"Stock Merger Consideration"**).

2.5. VPI shall provide to the Purchaser a listing of the issued and outstanding shares of the capital stock of VPI and list the VPI Shareholders by name. VPI shall identify to the Purchaser an allocation of the Stock Merger Consideration and the Cash Merger Consideration to be paid to each of the VPI Shareholders and officers of VPI. Such listing of the allocations of the Stock Merger Consideration and the Cash Merger Consideration from VPI shall be provided and detailed in **Schedule 2.5** attached hereto; and, the listing of the stock ownership as well as a listing of all officers and employees of VPI shall be provided and detailed in **Schedule 2.5(a)** attached hereto

2.6. Employment of Seller Principal. The consummation of the transactions contemplated by this Agreement is conditioned upon the Seller Principal agreeing to be employed by the Purchaser. At such time that Purchaser has sufficient revenues (in the sole discretion of Purchaser), the Seller Principal and the Purchaser shall enter into an Employment Agreement which will cover a minimal period of three years and which establishes the authority, duties and responsibilities as an executive in the Purchaser. Prior to the termination of this three-year period, the Parties agree to renegotiate in good faith the financial covenants of such employment for an additional three-year period. The Employment Agreement shall provide an initial annual salary of $190,000. In addition to salary, the Seller Principal shall be provided with 200,000 shares of the capital stock of Intelagents; which stock shall become vested 50,000 shares per year on each anniversary date of the date of this Agreement, for each of the next three years, provided that the Seller Principal remains employed by the Purchaser. In addition, Seller Principal shall use his best efforts to have Frank Inglis, the Chief Technical Officer of VPI, agree to enter into

an Employment Agreement with the Purchaser once the purchaser has sufficient revenues at an annual salary of $190,000.

3. Non-Competition Covenant.

All of VPI Shareholders agree, for a period of five (5) years following the Closing Date, not to compete or engage, directly or indirectly, in the business of the Purchaser to the extent Purchaser's business involves the business activities of VPI or Seller at the time of the Closing Date and/or involves business activities utilizing the assets purchased from Seller under this Agreement ("**Restricted Business Activities**"). This provision shall apply to the VPI Shareholders directly and shall include, but not be limited to, their activities in business as owners, partners or agents, or as employees of any person, firm, corporation or other entity engaged in such business, or in being interested directly or indirectly in any such business conducted by any person, firm, corporation or other entity. All such Seller hereto: (a) recognize that the value of the business of VPI and the Seller is based in large measure on the personal business relationships of the Seller and VPI; (b) acknowledge that a failure of the VPI Shareholders to comply with this covenant significantly diminishes the transactions contemplated by this Agreement; and, (c) agree that the provisions of this covenant shall survive the closing of this Agreement. The Non-competition Agreement to be executed by all of the VPI Shareholders is attached hereto as **Exhibit C**.

4. The Closing.

The closing of the transactions contemplated by this Agreement (the "**Closing**") will be held at the offices of Howard W. Carroll, P.C., 7250 North Cicero Avenue, Suite 201, Lincolnwood, Illinois, or at another place or under another acceptable process agreed to by the Parties, on _____, 2005, or any other date agreed to by the Parties. All required Schedules of this Agreement shall be provided by each Party to the other at least 5 business days before the Closing.

5. Representations and Warranties

5.1 Seller and VPI Representations and Warranties. The Seller, VPI and VPI Shareholders, jointly and severally, represent and warrant to the Purchaser as follows:

(a) Corporate Status. Each of Seller and VPI is a corporation duly organized and existing in good standing under the laws of the State of Delaware and is qualified to do business with full power, corporate and otherwise, to carry on its business and own its properties. Accurate and complete copies of the certificate of incorporation and bylaws of the Seller and VPI, including all amendments and certificates of good standing, are attached as **Schedule 5.1(a)**. Each of the Seller and VPI is qualified to do business and in good standing in all jurisdictions where the nature of its activities or ownership of properties requires such qualification and has duly filed all franchise and other tax returns required to be filed by the laws of such states and jurisdictions and has paid all taxes shown to be due and payable in such returns.

(b) Capitalization. The present equity capitalization of the Seller and VPI is as set forth in **Schedule 5.1(b)**. All outstanding capital stock of the Seller and VPI is validly issued, fully paid and non-assessable. Except as set forth in **Schedule 5.1(b)**, neither the Seller nor VPI has any authorized, issued or outstanding securities convertible into or exchangeable for capital stock, nor does any person or entity hold any option or right to purchase or otherwise acquire any shares of capital stock or any securities convertible into or exchangeable for such capital stock. Neither the Seller nor VPI has declared or has any outstanding commitments to pay, any dividend or to make any distribution or transfer of assets to its shareholders or persons affiliated with them, nor adopted or committed to any change in its equity capitalization.

(c) Title to Outstanding Shares. The shareholders of Seller and VPI, as listed, own all outstanding shares of Seller's and VPI's capital stock, free and clear of all pledges, liens, encumbrances, security interest, and claims whatsoever. Without limitation, none of the Seller, VPI nor their respective shareholders, officers, directors, agents, employees, or other representatives are parties to any existing agreement restricting the sale or transfer of the shares of Seller's or VPI's capital stock.

(d) Subsidiaries. Neither Seller nor VPI has any subsidiaries except those listed in **Schedule 5.1(d).** Except as set forth in **Schedule 5.1(d)**, the Seller or VPI, as applicable, owns all of the authorized and outstanding stock of any subsidiary free and clear of all pledges, liens, encumbrances, security interests, mortgages, deeds of trust, claims or restrictions.

(e) Corporate Authorization. The execution, delivery and performance by Seller and VPI of this Agreement and all other agreements and transactions contemplated herein have been duly authorized on the part of Seller and VPI by all requisite corporate action and will not violate or conflict with their respective certificates of incorporation or bylaws or with any law regulation, judgment, order, restriction or agreement to which they are subject.

(f) Consents. No consent, approval, permit, registration, filing or notice to or with any governmental agency or third party is required on the part of Seller or VPI for the execution, delivery and performance of this Agreement except as expressly provided herein.

(g) Financial Statements. (i) The financial statements of VPI for fiscal years ended December 31, 2003 and 2004 and for the nine (9) months ended September 30, 2005 are attached as **Schedule 5.1(g)** and reflect fairly the financial condition of VPI at such dates and their results of operations, retained earnings and changes in financial position for the fiscal years or shorter period then ended in accordance with generally accepted accounting principles consistently applied. All liabilities and obligations, existing and contingent, of VPI as of the date of this Agreement and as of the Closing Date are fairly reflected or disclosed and described in its financial statements and the notes thereto. Since the date of the last audit, VPI has conducted its activities only in the ordinary course and no material change in its financial conditions, results of operations or retained earnings occurred during such period. The net worth of VPI immediately prior to the consummation of the transactions contemplated by this Agreement will be no less than the net worth of VPI on December 31, 2004.

(h) Taxes. VPI has delivered to Purchaser complete and correct copies of the federal, state and local tax returns of VPI for each of the four years ended December 31, 2001, 2002, 2003 and 2004. Such returns and the information contained therein have been properly and accurately compiled and completed and accurately reflect the tax liabilities of VPI for the periods covered thereby. VPI has filed all federal, state and local tax returns which are required to be filed and has paid or made adequate provision for the payment of all taxes which have become or may become due with respect to operations during the periods to which such returns relate. VPI has delivered to Purchaser complete and correct copies of the reports of any audit of the income tax returns of VPI and of any deficiency letter and/or proposed assessment issued at the end of any such audit and all subsequent correspondence and documents relating thereto, except as disclosed in **Schedule 5.1(h).**

(i) Litigation, Proceedings or Claims. Except as described in **Schedule 5.1(i)** there is no litigation, governmental proceeding or investigation, or claim pending or threatened against Seller or VPI or their respective properties or business or against or relating to the transactions contemplated by this Agreement.

(j) Description of Properties, Contracts and Material Information. **Schedule 5.1(j)** is an accurate and complete list as of the date of this Agreement of: (i) all real property and all items of equipment presently owned or leased by the Seller or VPI with a brief description of each property and its use, copies of title instruments and leases, and details relating to any liens, encumbrances or claims thereto and any direct or indirect interest therein of any of the Seller's or VPI's directors or officers; (ii) all patents, trademarks, trade names, copyrights and intellectual properties including all registration thereof and applications therefore presently owned in whole or in part by the Seller or VPI, and all patent, trademark or copyright and intellectual property licenses to which the Seller or VPI is a party; (iii) all bonds, debentures, notes, stock or other securities other than stock of subsidiaries already listed in a schedule hereto, and all accounts receivable other than trade accounts receivable which are not more than 90 days old, held or owned by the Seller or VPI; (iv) all policies of insurance in force covering or owned by the Seller or VPI; (v) all loan agreements or bank credit agreements in effect, setting forth the amount of the original loan, the unpaid balance, the interest rate and payments, the maturity date, any prepayment penalties and the name of the lender; (vi) all material agreements to which the Seller or VPI is a party except that the Seller and VPI may omit agreements made in the ordinary course of business which are terminable by the Seller or VPI by notice of not more than 90 days without penalty or which do not obligate the Seller or VPI in amounts in excess of $25,000 in the aggregate per agreement; (vii) all employment contracts with any officers and employees whose current annual salary rate is U.S. $50,000 or more, together with a description of all incentive, compensation, bonus, profit sharing retirement, pension or other similar plans or arrangements for any of such officers or employees and (viii) all agents, consultants and independent contractors retained by Seller or VPI, with a brief description of the arrangement for compensation, and all persons, if any, holding a power of attorney, to act on their behalf (ix) all agreements and transactions entered into and in force with any officer, director or stockholder of the Seller or VPI or any person related to any of them, except for agreements which are terminable by the Seller or VPI within 90 days without penalty or which do not obligate the Seller or VPI in amounts in excess of $25,000 in the aggregate per agreement. Complete and correct copies of all agreements, instruments or other documents relating to the items referred to

above are provided and disclosed in Schedule 6.1(j) and have been delivered or made available to the Purchaser. Neither Seller nor VPI, nor to the knowledge of Seller and VPI any other party thereto, is in material default on any obligation to be performed by it under any contract described in this section, or in material violation of any law, ordinance, regulation, order or decree applicable to it.

(k) Legal Compliance. Seller and VPI are in compliance with all laws, regulations, permits and orders applicable to their respective business and assets. Neither Seller nor VPI is infringing any patent, trademark or copyright or intellectual property or using or disclosing without authorization any trade secret of third parties except as disclosed in **Schedule 5.1(k)**.

(l) Employee Relations. Since January 1, 2002, VPI has not experienced any work interruptions, organization campaigns or other concerted actions by its employees and has not received any complaints of failure to comply with equal employment opportunity law except as disclosed in **Schedule 5.1(l)**.

(m) Employee Benefit Plans. All employee benefit plans, as defined in the Employee Retirement Income Security Act of 1974, as amended, covering present and former employees of VPI have been fully disclosed in **Schedule 5.1(m)**, including, without limitation, all commitments to provide employee benefits. Any plans intended to be qualified plans and trusts intended to be exempt organizations under the Internal Revenue Code are qualified and exempt and VPI has determination letters evidencing such status. There has been, and is, no reportable event, accumulated funding deficiency, termination liability, withdrawal liability or prohibited transaction in connection with such plans. VPI has no obligations to provide post-retirement health, medical, death or other welfare benefits. All group health plans have been maintained in compliance with the Internal Revenue Code. All vacation, severance and similar plans or policies of VPI have been fully and accurately disclosed in the foregoing schedules.

(n) Real Property. Neither Seller nor VPI owns or leases any real property.

(o) Broker's Fees. Neither Seller nor VPI has any liability or obligation to pay any fees or commissions to any broker, finder, or agent with respect to the transactions contemplated by this Agreement.

6. **Representations and Warranties of Purchaser.**

The Purchaser and Intelagents, jointly and severally, represent and warrant to the Seller and VPI as follows:

(a) Corporate Status. Each of Purchaser and Intelagents is a corporation duly authorized and existing in good standing under the laws of the State of Delaware with full corporate power and authority to carry out the transactions contemplated by this Agreement.

(b) Corporate Authorization. The execution, delivery and performance by Purchaser and Intelagents of this Agreement and all other agreements and transactions

contemplated herein have been duly authorized on the part of Purchaser and Intelagents by all requisite corporate action and will not violate or conflict with their respective certificates of incorporation or bylaws or with any law regulation, judgment, order, restriction or agreement to which they are subject.

(c) Consents. No consent, approval, permit, registration, filing or notice to or with any governmental agency or third party is required on the part of Purchaser or Intelagents for the execution, delivery and performance of this Agreement except as expressly provided herein.

(d) Broker's Fees. Neither Purchaser nor Intelagents has any liability or obligation to pay any fees or commissions to any broker, finder, or agent with respect to the transactions contemplated by this Agreement.

7. Conditions Precedent to the Obligations of the Parties.

7.1 Conditions to the Obligations of Purchaser. The obligation of Purchaser and Intelagents to close the transactions contemplated by this Agreement is subject to the following conditions:

(a) Representations and Warranties. The representations and warranties of Seller and VPI shall continue to be true and correct in all material respects on the Closing Date and Seller and VPI shall have performed all covenants and other obligations, including payment of the Merger Consideration, required of them under this Agreement.

(b) No Material Adverse Change. There shall have been no material adverse change in Seller's or VPI's business, assets, financial condition or results of operations.

(c) Delivery of Certain Documents. Seller and VPI shall have delivered to Purchaser and Intelagents at the Closing all of the documents described and requested in this Agreement.

(d) The Purchaser shall have satisfactorily completed its due diligence investigation of the Seller and VPI.

(e) VPI shall have filed a certificate of amendment to VPI's Certificate of Incorporation with the Delaware Secretary of State to change its name to such name other than "Verify and Protect, Inc." or any similar or confusing name as shall be reasonably satisfactory to Purchaser and Purchaser shall have the right to use the name "Verify and Protect, Inc." and other similar names from and after the Closing.

(f) All of the assets of VPI set forth on VPI's financial statements for the nine (9) months ended September 30, 2005 shall have been contributed by VPI to Seller and specific liabilities of VPI disclosed herein shall have been assumed by Seller.

(g) VPI shall be liquidated promptly following the Closing.

7.2 Conditions to the Obligations of the Seller and VPI. The obligations of the Seller and VPI to close the transactions contemplated by this Agreement are subject to the following conditions:

(a) Representations and Warranties. The representations and warranties of Purchaser and Intelagents shall continue to be true and correct in all material respects on the Closing Date and Purchaser and Intelagents shall have performed all covenants and other obligations required of them under this Agreement.

8. Documents to be delivered at Closing.

8.1 Documents Delivered by the Seller and VPI. At the Closing, the Seller and VPI shall deliver to the Purchaser the following documents:

(a) Such evidence of corporate existence, qualification, good standing, incumbency of officers, adoption of resolutions and evidence of other corporate procedures and authority as may reasonably be requested by counsel for the Purchaser.

(b) Evidence that any property transfers cited in any of the Schedules have been made.

(c) A certificate signed by the Seller and VPI updating and reaffirming the representations and warranties set forth in this Agreement and confirming performance of all the covenants set forth in this Agreement to the extent they are to be performed by them on or before the Closing Date.

(d) "Comfort Letters and current statements" from _____, the certified public accountants of VPI dated the Closing Date, in the form attached as **Schedule 8.1(d)**.

(e) A release signed by the Seller, VPI and Seller Principal in the form attached as **Schedule 8.1(f)**.

(f) Certificates representing all issued and outstanding shares of Common stock of VPI held by the VPI Shareholders together with stock powers endorsed in blank with the signature guarantees of a bank included in **Schedule 10.1(f)**.

(g) Non-competition Agreements executed by each of the VPI Shareholders.

8.2 Documents to be delivered by Purchaser and Intelagents. At the Closing and upon fulfillment of the conditions described in this Agreement, the Purchaser and Intelagents shall deliver documents and make payments as follows:

(a) Such evidence of corporate existence, qualification, good standing, incumbency of officers, adoption of resolutions and evidence of other corporate procedures and authority as may reasonably be requested by counsel for the Seller.

(b) A certificate signed by an officer of the Purchaser updating and reaffirming its representations and warranties set forth in this Agreement and confirming performance of all the covenants set forth in this Agreement to the extent they are to be performed by the Purchaser on or before the Closing Date.

(c) The Promissory Note.

(d) The Stock Merger Consideration.

8.3 Simultaneous Delivery. All documents delivered at the Closing shall be deemed to have been delivered simultaneously.

9. Indemnification

9.1 Agreement by Seller, VPI and the VPI Shareholders to Indemnify. Seller, VPI and the VPI Shareholders, jointly and severally, hereby indemnify, protect, defend and forever hold Purchaser, Intelagents and their assignees, shareholders, officers and employees harmless in respect of the aggregate of all indemnifiable damages of Purchaser and Intelagents. For this purpose, "indemnifiable damages" of Purchaser and Intelagents means the aggregate of all fees, expenses, losses, costs, deficiencies, liabilities and damages (including reasonable attorney fees and expenses) incurred or suffered by Purchaser or Intelagents (a) resulting from any inaccurate representation or warranty made by Seller or VPI hereunder, or (b) resulting from any default in the performance of any of the covenants or agreements made by Seller or VPI in this Agreement. Purchaser shall have the right to set off any indemnifiable damages of Purchaser and Intelagents against the Promissory Note.

9.2 Agreements by Purchaser and Intelagents to Indemnify. Purchaser and Intelagents, jointly and severally, hereby indemnify, protect, defend and forever hold Seller, VPI and the VPI Shareholders harmless in respect of the aggregate of all indemnifiable damages of Seller, VPI and the VPI Shareholders. For this purpose, "indemnifiable damages" of Seller, VPI and the VPI Shareholders means the aggregate of all fees, expenses, losses, costs, deficiencies, liabilities and damages (including reasonable attorney fees and expenses) incurred or suffered by Seller, VPI and the VPI Shareholders resulting from (a) any inaccurate representation or warranty made by Purchaser or Intelagents hereunder or (b) resulting from any default in the performance of any of the covenants or agreements made by Purchaser or Intelagents in this Agreement.

9.3 Matters Involving Third Parties. If any third party shall notify any party (the "**Indemnified Party**") with respect to any matter which may give rise to a claim for indemnification against any other Party (the "**Indemnifying Party**") under this Section 10, then the Indemnified Party shall notify each Indemnifying Party thereof promptly; provided, however, that no delay on the part of the Indemnified Party in notifying any Indemnifying Party shall relieve the Indemnifying Party from any liability or obligation hereunder unless (and then solely

to the extent) the Indemnifying Party thereby is actually and directly damaged. In the event any Indemnifying Party notifies the Indemnified Party within 15 days after the Indemnified Party has given notice of the matter that the Indemnifying Party is assuming the defense thereof, (a) the Indemnifying Party will defend the Indemnified Party against the matter with counsel of its choice satisfactory to the Indemnified Party, (b) the Indemnified Party may retain separate co-counsel at its sole cost and expense (except that the Indemnifying Party will be responsible for the fees and expenses of the separate co-counsel to the extent the Indemnified Party concludes that the counsel the Indemnifying Party has selected is, in its reasonable belief, inadequate or will be ineffective or has a conflict of interest), (c) the Indemnified Party will not consent to the entry of any judgment or enter into any settlement with respect to the matter without the written consent of the Indemnifying Party (not to be withheld or delayed unreasonably), and (d) the Indemnifying Party will not consent to the entry of any judgment with respect to the matter, or enter into any settlement which does not include a provision whereby the plaintiff or claimant in the matter releases the Indemnified Party from all liability with respect thereto, without the written consent of the Indemnified Party (not to be withheld or delayed unreasonably). In the event no Indemnifying Party notifies the Indemnified Party within 15 days after the Indemnified Party has given notice of the matter that the Indemnifying Party is assuming the defense thereof, however, the Indemnified Party may defend against, or enter into any settlement with respect to, the matter in any manner it may deem appropriate.

9.4 Survival of Representations and Warranties. The representations and warranties of Seller and VPI contained in Sections 6.1(a), 6.1(b), 6.1(c), 6.1(e), 6.1(f), 6.1(h) and 6.1(i) shall survive the Closing hereunder and continue in full force and effect forever thereafter, subject to any applicable statutes of limitations. All of the other representations and warranties of the Parties contained in this Agreement shall survive the Closing and continue in full force and effect thereafter for a period of eighteen (18) months.

10. General Provisions.

10.1 Further Actions. The Parties agree to execute and deliver from time to time hereafter any and all such further documents and to take such further actions as shall be reasonably necessary to carry out the transactions contemplated by this Agreement.

10.2 Non-Assignability. Neither this Agreement nor any rights hereunder may be assigned or otherwise transferred directly or indirectly by any Party without the prior written consent of the other Parties and any attempt to do so shall be null and void, provided that this Agreement and the rights and obligations herein shall inure to the benefit of, and be binding upon, the executors, administrators, heirs and successors of the Parties.

10.3 Entire Agreement. In entering into and closing this Agreement, no Party has relied or shall rely upon any promises, representations and warranties not expressed herein, and this Agreement expresses their entire agreement on the subject matter.

10.4 Amendment and Waiver. Neither this Agreement nor any provision or provisions herein may be amended or waived except by a written amendment or new agreement executed by all the Parties.

10.5 Governing Law. The validity, interpretation, performance and enforcement of this Agreement shall be governed by the laws of the State of Illinois without regard to the application of its conflict of laws rules. The Parties agree that the exclusive venue for any disputes, actions or conflicts with respect to this Agreement shall be the Illinois state or United States federal courts located in Chicago, Illinois, and each Party waives its right to trial by jury.

10.6 Notices. All notices or other communications hereunder shall be given in writing and shall be deemed to be, if duly given if delivered or mailed, first class postage prepaid, to the following addresses:

To the Seller or VPI:

Copy to:

To the Purchaser
or Intelagents: Gregory E. Webb, Chairman and CEO
304 E. Fairview Street
Arlington Heights, IL 60005

Copy to: Howard W. Carroll, Esq.
Howard W. Carroll, P.C.
7250 N. Cicero Ave., Suite 201
Lincolnwood, IL 60712

10.7 Expenses. Each Party shall pay all costs and expenses incurred by it (including, without limitation, the payment of all fees and expenses of its counsel) in carrying out its respective obligations under this Agreement and the transactions contemplated herein.

[SIGNATURE PAGE FOLLOWS]

IN WITNESS WHEREOF, the parties have executed and delivered this Agreement as of the _____ day of _____, 2005.

VPI ACQUISITION, INC.

By: _____
Gregory E. Webb, Chairman and CEO

INTELAGENTS, INC.

By: _____
Gregory E. Webb, Chairman and CEO

VERIFY AND PROTECT, INC.

By:_____
Jeffry Boston, CEO

VPI TARGET , INC.

By:_____
Jeffry Boston, CEO

VPI SHAREHOLDERS

_____
Jeffry Boston

_____
[ ]

_____
[ ]

EXHIBIT A

PROMISSORY NOTE

EXHIBIT B

NON-COMPETITION AGREEMENT

CYBERDRIVEILLINOIS

JESSE WHITE
SECRETARY OF STATE

SERVICES PROGRAMS PRESS PUBLICATIONS DEPARTMENTS CONTACT

CORPORATION FILE DETAIL REPORT

Entity Name	INFRAEGIS, INC.	**File Number**	62953381
Status	ACTIVE		
Entity Type	CORPORATION	**Type of Corp**	FOREIGN BCA
Qualification Date (Foreign)	06/20/2003	**State**	DELAWARE
Agent Name	ILLINOIS CORPORATION SERVICE C	**Agent Change Date**	07/25/2011
Agent Street Address	801 ADLAI STEVENSON DRIVE	**President Name & Address**	GREGORY E WEBB 1612 LANDMEIER RD UNIT F ELK GROVE IL 60007
Agent City	SPRINGFIELD	**Secretary Name & Address**	CHARLES R ABBOTT SAME
Agent Zip	62703	**Duration Date**	PERPETUAL
Annual Report Filing Date	07/25/2011	**For Year**	2011
Old Corp Name	07/18/2006 - INTELAGENTS, INC. 07/18/2006 - INFRAAEGIS, INC.		

Return to the Search Screen

Purchase Certificate of Good Standing

(One Certificate per Transaction)

BACK TO CYBERDRIVEILLINOIS.COM HOME PAGE

GENERAL AND MUTUAL RELEASE

THIS GENERAL AND MUTUAL RELEASE *("Release")* is made this 28th day of April, 2008 by and between INFRAEGIS, INC. ("IA") a Delaware company, with offices located at 1612 Landmeier Road, Elk Grove Village, Illinois 60007, VERIFY AND PROTECT, INC. *("VPI")* a Delaware company, with offices located at 3333 Warrenville Road, Suite 200, Lisle, Illinois 60532, and Michael Lang, Chris Lang, and Jeffrey Boston, all certain shareholders of VPI (collectively, the *"VPI Shareholders")* (the VPI Shareholders and VPI are collectively referred to below as the *"VPI Parties").*

RECITALS:

WHEREAS, on November 21, 2005, VPI and IA entered into an agreement (the *"Prior Merger Agreement")* that contemplated that VPI would be merged into IA pursuant to terms and conditions provided for therein (the *"Proposed Merger")*;

WHEREAS, on or about November 21, 2005, IA paid $500,000.00 and further delivered to VPI a promissory note in the principal amount of $2,250,000.00 (the *"Proposed Merger Note")* in furtherance of the Proposed Merger;

WHEREAS, VPI and IA thereafter failed to complete the Proposed Merger as certain conditions were never satisfied and no certificate of merger has been filed by VPI or IA, the parties have previously represented and agreed that the Proposed Merger had been terminated, and the parties desire to more formally terminate and cancel the Prior Merger Agreement and Proposed Merger, and all other prior agreements by and between IA and the VPI Parties, upon the terms and conditions stated herein;

WHEREAS, IA has claimed that VPI collected for itself $53,000 which was to be paid to IA, and which IA has requested be paid to IA *("IA Claim")*;

WHEREAS, VPI claims to be in the business of developing, designing and producing user authentication and data encryption software;

WHEREAS, VPI claims that it designed, produced and patented DeepCloak™, a self-contained technology that authenticates users, generates all keys dynamically, encrypts data and creates digital signatures without use of digital certificates or third party certificate authorities (collectively the *"VPI Products")*;

WHEREAS, VPI claims that the VPI Products are designed and produced to provide software developers, device makers and corporation IT departments the best means to secure their applications with the DeepCloak™ technology;

WHEREAS, IA claims to be in the business of providing advanced wired and wireless products and services for protection of fixed and mobile assets against theft, tampering or terrorist attacks using nuclear, chemical or biological materials;

WHEREAS, the parties hereto wish to resolve all disputes between themselves, without any admission of liability, including but not limited to the IA Claim;

NOW, THEREFORE, in consideration of the recitals and mutual promises contained herein and other good and valuable consideration, the receipt and sufficiency of which are hereby acknowledged, it is agreed by and between IA and the VPI Parties as follows:

1. *Recitals.* The foregoing recitals are incorporated as an integral part of this Agreement.

2. *Termination and Cancellation of Prior Merger Agreement and Proposed Merger.* To the extent not otherwise formally terminated and canceled prior to this Agreement, the Prior Merger Agreement and Proposed Merger, the Proposed Note, and all other agreements by and between IA and the VPI Parties, are hereby terminated and canceled, and any obligations or rights owed pursuant to such agreements are hereby released to the fullest extent possible.

3. *Return of all VPI Products.* The IA Releasing Parties (as defined below) shall, upon execution of this Agreement, immediately return to VPI all of the VPI Products in their possession or control (exclusive of any VPI Products in the possession or control of IA's former attorneys, Carroll & Sain), together with a representation that all such VPI Products have been returned to VPI in accord with this Agreement and that the IA Releasing Parties do not claim any interest in or right to utilize any of the VPI Products. VPI hereby grants a fully paid license to IA covering any use by IA of the VPI Products prior to the execution of this Agreement.

4. *Releases by IA Releasing Parties.* IA, on behalf of itself and all of its affiliates, predecessors, successors and assigns, respective past and present officers, directors, shareholders, partners, members, managing members, employees, agents, representatives, administrators, attorneys, insurers, fiduciaries, heirs, and executors, in their individual and representative capacities *("IA Releasing Parties"),* forever release VPI, the VPI Shareholders, and all of their past and present officers, directors, shareholders, partners, members, managing members, employees, agents, representatives, administrators, attorneys, insurers, fiduciaries, heirs, and executors, in their individual and representative capacities **("VPI** *Released Parties")* from any and all causes of action, agreements, damages, judgments, claims, debts, covenants, executions and demands of any kind whatsoever, which the IA Releasing Parties ever had, now have or may have against the VPI Released Parties, or any *of* them, in law or equity, whether known or unknown, for, upon, or by reason *of,* any matter whatsoever occurring up to the date of this Agreement, including without limitation any matter in connection with or in relation to: (i) the Proposed Merger, Prior Merger Agreement; and Proposed Merger Note received by VPI (ii) the IA claim; and (iii) any other matter between any parties to this Agreement. IA and the VPI Parties intend for this release to be construed as a general release.

5. *Releases by VPI Released Parties.* The VPI Released Parties hereby forever release the IA Releasing Parties from any and all causes of action, agreements, damages, judgments, claims, debts, covenants, executions and demands of any kind whatsoever, which the VPI Released Parties had, now have or may have against the IA Releasing Parties or any of them, in law or equity, whether known or unknown, for, upon, or by reason of, any matter

whatsoever occurring up to the date of this Agreement, including without limitation any matter in connection with or in relation to: (i) the Proposed Merger, Prior Merger Agreement, and Proposed Merger Note received by VPI; (ii) claims involving any investments by the VPI Released Parties in IA; and (iii) any other matter between any parties to this Agreement. IA and the VPI Parties intend for this release to be construed as a general release. Further, Boston represents and warrants that he is not entitled to any shares, warrants, or other interests in IA pursuant to any agreement with IA and that any such agreement between IA and Boston is not of any force and effect, and Boston forever releases any claim of right relating to any warrants to purchase IA stock.

6. *Nondisparagment.* IA and the VPI Parties agree that from and after the date of this Agreement, they will not make or solicit or encourage others to make any untrue allegations, statements or remarks, either oral or written, concerning any of the parties hereto. The parties agree that they and their respective successors, assigns and attorneys shall keep the terms of this Agreement confidential except to the extent any Party makes disclosures to its attorneys or tax advisors, any Party seeks to enforce this Agreement. If asked, the Parties shall state that the matters between then have been resolved. IA and the VPI Parties also agree to forever refrain from directly or indirectly (a) disparaging each other, or their officers, directors, employees, agents, attorneys, representatives, or related entities; or (b) encouraging, voluntarily participating in or, cooperating with any other party in any lawsuit, investigation, claim or action involving any other party to this agreement unless required to do so by law. Nothing in this provision shall preclude any claim that may arise by virtue of a breach of an undertaking or promise set forth in this Agreement.

7. *Dismissal With Prejudice All Actions Brought by IA.* Upon execution of this Agreement, IA shall take all action necessary to cause an immediate dismissal, with prejudice, of Michael Lang from those actions currently pending in Delaware Chancery Court, Civil Action Nos. 2235-N and 2229-N. IA further represents and warrants that it has not brought any other actions that are currently pending against Michael Lang, VPI, or any of the other VPI Released Parties.

8. *Voluntary and Knowing Execution.* The IA Releasing Parties and the VPI Parties represent that they have read and fully understand the terms of this Agreement, and that they had had the opportunity to consult with an attorney prior to signing this Agreement. The IA Releasing Parties and the VPI Released Parties acknowledge that they are executing this Agreement voluntarily and knowingly and have not relied on any representations, promises or agreements of any kind made in connection with the parties' decision to accept the terms of this Agreement, other than those set forth in this Agreement. The parties also represent that each has the requisite authority to enter into this Agreement.

9. *Notices.* All notices related to this Agreement shall be made in writing as follows: (a) by actual delivery into the hands of the party entitled to receive it or by facsimile to such party, in which case the notice shall be deemed given on the date it is sent; or (b) by, Federal Express or any other carrier, in which case the notice shall be deemed given on the second day following the date it is deposited with such carrier. All notices provided under this Agreement shall be to the last known address of the party entitled to receive it. Any party to this Agreement

may change its address for notice purposes, by providing written notice of the change of address to each of the other parties. All notices under this Agreement shall be addressed as follows:

(a) if to IA Releasing Parties: InfrAegis, Inc.
1612 Landmeier Road
Suite F
Elk Grove Village, IL 60007

Attention: James R. Zilka, CFO

(b) if to VPI Released Parties, to: VPI Inc.
3333 Warrenville Road, Suite 200
Lisle, IL 60532

with a copy to: David T. B. Audley
Chapman and Cutler LLP
111 West Monroe Street
Chicago, IL 60603

10. *Governing Law.* The laws of the State of Illinois (other than those pertaining to conflicts of law) shall govern all aspects of this Agreement, irrespective of the fact that one of the parties now is or may become a resident of a different state or country. The parties shall submit all disputes which arise under this Agreement to state or federal courts located in Illinois for resolution. Each party irrevocably submits itself to the jurisdiction of said courts for the purpose of all said disputes. The parties acknowledge the aforesaid courts shall have exclusive jurisdiction over this Agreement, and specifically waive and agree not to assert by way of motion or as a defense any claims which they may have that involve jurisdiction or venue, including but not limited to forum non conveniens. Service of process for any claim which arises under this Agreement shall be valid if mailed to the party being served, by first-class mail, Federal Express, UPS, or another carrier. If service of process is made as aforesaid, the party served agrees that such service shall constitute valid service, and specifically waives any objections the party served may have under any state or federal law or rule concerning service of process. Service of process in accordance with this section shall be in addition to and not to the exclusion of any other service of process method legally available.

11. *No Assignment.*: The parties hereto represent and warrant that they have not previously assigned any rights or duties identified herein, or any of the claims being released hereby.

12. *No Other Agreements.* This Agreement constitutes the entire understanding and agreement of the parties as to the matters set forth in this Agreement. No alteration of or amendment to this Agreement shall be effective unless given in writing and signed by the party or parties sought to be charged or bound by the alteration or amendment.

13. *Counterparts.* This Release may be signed in counterparts with the same effect as if the signatures hereto were upon the same instrument.

IN WITNESS WHEREOF, the parties have executed and delivered this Settlement Agreement as of the date first stated above.

INFRAEGIS INC.

By _____
Gregory E. Webb, Chairman, CEO and President

VERIFY AND PROTECT, INC.

By _____
Michael Lang, individually and as Chairman, CEO and President of Verify and Protect, Inc.

By _____
Chris Lang, individually as a shareholder of Verify and Protect, Inc.

By _____
Jeffrey Boston, individually as a shareholder of Verify and Protect, Inc.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent No.: **6,856,975**

Applicant: **Frank Inglis**

Assignee: **Verify & Protect, Inc.**

Filed: **March 30, 2000**

Granted: **February 15, 2005**

OFFICE OF PETITIONS

Title: **SYSTEM, METHOD, AND ARTICLE OF MANUFACTURE FOR SECURE TRANSACTIONS UTILIZING A COMPUTER NETWORK**

April 16, 2012

FILED ELECTRONICALLY FILED VIA EFS-WEB

Declaration of Jeffry K. Boston

1. I, Jeffry Boston, hereby state and affirm the following:

2. I am a Director and manager of Verify and Protect, Inc. ("VPI"), the assignee of all of the rights and interest to U.S. Patent No. 6,856,975 (the "'975 patent"). (copy of Assignment attached as Exhibit 1). During the time period of 2000 through 2005 I served as President of VPI.

3. On March 30, 2000 the Halvorson Law Firm, on behalf of inventor Frank Inglis and, after assignment, VPI filed U.S. Patent Application Serial No. 09/540,193 (the "'193 application"). At the time of the application Frank Inglis was contracted to VPI and, pursuant to that contract relationship and an Assignment By Inventor of Patents executed by Mr. Inglis, he assigned all right, title and interest to the '193 application to VPI.

4. From March of 2000 through February of 2005, I worked closely with the inventor, Mr. Inglis, and the Halvorson Law firm, via both telephone and electronic mail, to formulate responses to Office Actions from the U.S. Patent & Trademark Office ("PTO").

5. On February 15, 2005, the '193 application issued as the '975 patent (copy of '975 patent attached as Exhibit 2). As a manager of VPI, I understood that certain fees were to be paid in connection with the patent application and issued patent as required by the PTO. To ensure timely payment of all fees, VPI relied on the Halvorson Law Firm to correctly calendar the maintenance fee due dates on their docketing system and provide notice when payments were due. The Halvorson Law Firm took steps to maintain current contact information at the PTO as attorneys of record for the '975 patent. (Copy of docket sheet showing that the Halvorson Law Firm updated and confirmed its contact address when the issue fee was paid attached as Exhibit 3).

6. After a period of time during which various assets, including the '975 patent, were transferred to the control of a third party, VPI reacquired control of the assets in 2008. The principal asset of VPI as of 2008 was the '975 patent.

7. During the time period described above, and at all times since the filing of the '193 application, VPI continued to rely on the Halvorson Law Firm and its docketing system to provide notice of deadlines related to the '975 patent.

8. In November 2011, VPI began to take steps to develop, implement, and practice the invention of the '975 patent. In connection with those efforts, we engaged counsel to advise on intellectual property rights.

9. After engaging counsel in November 2011, I learned that the four-year maintenance fee payment for the '975 patent had become due but had not been paid. I have since learned that the four-year maintenance fee payment for the '975 patent was due by February 15, 2009.

10. VPI was not aware of the due date for the Maintenance Fee and was relying upon its patent counsel, the Halvorson Law Firm, to notify VPI of the fees that were due to the PTO in connection with preserving the patent rights for the '975 patent. VPI never received any form of notice from the Halvorson Law Firm, or any other source, that a maintenance fee payment was due.

11. As of February 15, 2009, VPI had not received any notice from the Halvorson Law Firm, or any other source, that a maintenance fee payment was due.

12. VPI did not receive a notice of abandonment from the PTO or from the Halvorson Law Firm.

13. VPI first realized that the '975 patent had been abandoned in November 2011 and took prompt steps to revive it by retaining the law firm of SNR Denton US LLP to protect its rights to the '975 patent.

14. All actions taken by VPI with respect to the '975 patent have been reasonable and calculated to avoid non-payment of fees.

15. Non-payment of the four-year maintenance fee was unavoidable due to the reliance of VPI on its patent counsel at the Halvorson Law Firm.

17. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true and further that the

statements are made with the knowledge that willful statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements jeopardize the validity of the application or any patent issuing thereon.

April 16, 2012

Jeffry K. Boston
Manager, Verify and Protect, Inc.

EXHIBIT A

ASSIGNMENT BY INVENTOR OF PATENTS

For good and valuable consideration, the receipt of which is hereby acknowledged, the below signed inventor, a citizen of the United States of America, residing at the addresses listed below his name, sells and assigns to Verify and Protect, Inc., a Delaware Corporation having a place of business at 3333 Warrenville Road, Suite 200, Lisle IL 60532, its successors and assigns, all their right, title and interest in and to the following patents and patent application:

(1) U.S. Patent Application, Currently Entitled "System, Method, and Article of Manufacture for Secure Transactions Utilizing a Computer Network", Ser. No. 09/540,193, filed on 03/30/2000.

and all inventions contained therein, all improvements thereon, all technologies related thereto, all reissues and extensions thereof, and covenant that he has full right so to do, and agrees that he will communicate to said Corporation or its representatives any facts known to him respecting said improvements and testify in any legal proceeding, sign all lawful papers, execute all, reissue and extension applications, make all rightful oaths, and generally do everything possible to aid said Corporation, its successors, assigns and nominees, to obtain and enforce patent protection for said inventions in all countries.

IN TESTIMONY WHEREOF, I hereunto set my hand and seal this 15th day of OCT, 2002.

Frank H. Inglis
3215 E. Muirwood Dr.
Phoenix, Arizona 85044

State of Arizona ss. County of Maricopa

On this 15th day of Oct, 2002 before me, a Notary Public in and for the State and County aforesaid, personally appeared and to me known be the person of the above signed names, who signed and sealed the foregoing instrument, and they acknowledged the same to be their free act and deed.

Notary Public

(Seal)

OFFICIAL SEAL
LISA M. BIONDI
NOTARY PUBLIC-ARIZONA
MARICOPA COUNTY
MY COMM. EXPIRES DEC. 19, 2005

US006856975B1

(12) **United States Patent**
Inglis

(10) **Patent No.: US 6,856,975 B1**
(45) **Date of Patent: Feb. 15, 2005**

(54) **SYSTEM, METHOD, AND ARTICLE OF MANUFACTURE FOR SECURE TRANSACTIONS UTILIZING A COMPUTER NETWORK**

(75) Inventor: **Frank Inglis**, Phoenix, AZ (US)

(73) Assignee: **Verify & Protect Inc.**, Lisle, IL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/540,193**

(22) Filed: **Mar. 30, 2000**

(51) **Int. Cl.**[7] **G06F 17/00**

(52) **U.S. Cl.** **705/51**; 705/1; 705/40; 705/54

(58) **Field of Search** 705/51, 54, 1

(56) **References Cited**

PUBLICATIONS

Spring, E-Business Security Technologies, 2001.*

* cited by examiner

Primary Examiner—Richard Weisberger
(74) *Attorney, Agent, or Firm*—The Halvorson Law Firm

(57) **ABSTRACT**

The present invention is a system or method and device useful for the secure electronic payment of consumer debts over a publicly accessible computer network. The preferred form of the present invention uses at least two separate, but compatible, software packages. Security server software that continuously runs on a security server and payor software that runs on demand on a payor computer system. The payor computer system communicates via the payor software with the security server via the security server software. The communication, or transaction, session operates under the secure communication protocol described below. A payee computer system may also communicate via payee software with the security server. Additionally, a version is provided that utilizes smart card technology and a remote kiosk computer that communicates with the security server.

10 Claims, 6 Drawing Sheets

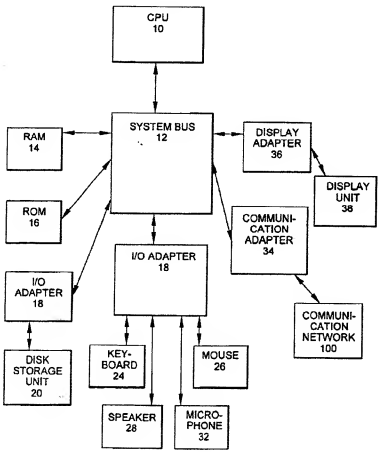


Fig. 1

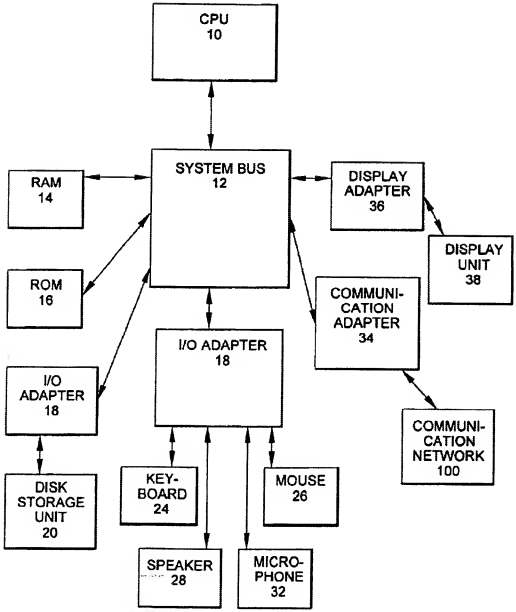

CPU
10
SYSTEM BUS
12
RAM
14
ROM
16
I/O
ADAPTER
18
I/O ADAPTER
18
DISK
STORAGE
UNIT
20
DISPLAY
ADAPTER
36
DISPLAY
UNIT
38
COMMUNI-
CATION
ADAPTER
34
COMMUNI-
CATION
NETWORK
100
KEY-
BOARD
24
MOUSE
26
SPEAKER
28
MICRO-
PHONE
32

Fig. 2

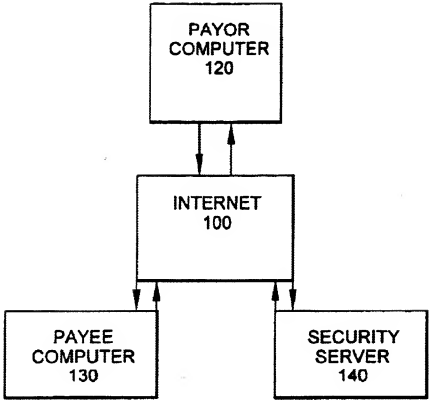

PAYOR
COMPUTER
120
INTERNET
100
PAYEE
COMPUTER
130
SECURITY
SERVER
140

Fig. 3A

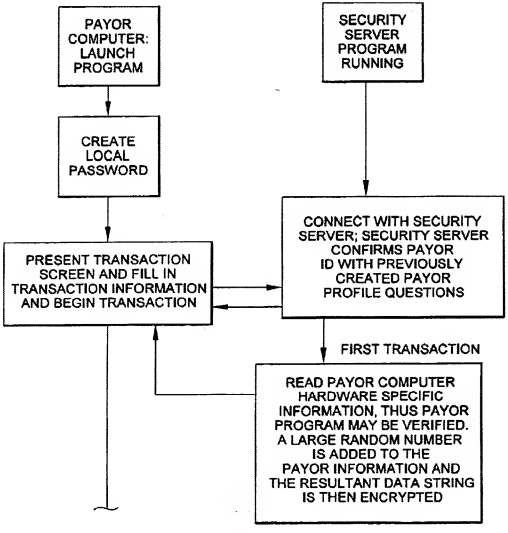

PAYOR COMPUTER: LAUNCH PROGRAM
SECURITY SERVER PROGRAM RUNNING
CREATE LOCAL PASSWORD
CONNECT WITH SECURITY SERVER; SECURITY SERVER CONFIRMS PAYOR ID WITH PREVIOUSLY CREATED PAYOR PROFILE QUESTIONS
PRESENT TRANSACTION SCREEN AND FILL IN TRANSACTION INFORMATION AND BEGIN TRANSACTION
FIRST TRANSACTION
READ PAYOR COMPUTER HARDWARE SPECIFIC INFORMATION, THUS PAYOR PROGRAM MAY BE VERIFIED. A LARGE RANDOM NUMBER IS ADDED TO THE PAYOR INFORMATION AND THE RESULTANT DATA STRING IS THEN ENCRYPTED

Fig. 3B

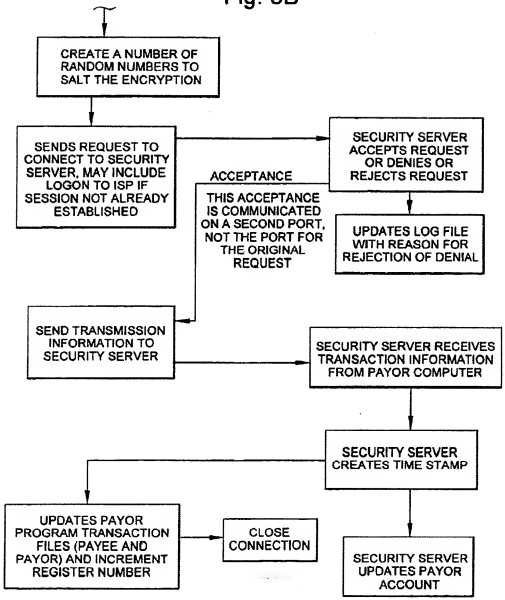

CREATE A NUMBER OF RANDOM NUMBERS TO SALT THE ENCRYPTION
SENDS REQUEST TO CONNECT TO SECURITY SERVER, MAY INCLUDE LOGON TO ISP IF SESSION NOT ALREADY ESTABLISHED
SECURITY SERVER ACCEPTS REQUEST OR DENIES OR REJECTS REQUEST
ACCEPTANCE
THIS ACCEPTANCE IS COMMUNICATED ON A SECOND PORT, NOT THE PORT FOR THE ORIGINAL REQUEST
UPDATES LOG FILE WITH REASON FOR REJECTION OF DENIAL
SEND TRANSMISSION INFORMATION TO SECURITY SERVER
SECURITY SERVER RECEIVES TRANSACTION INFORMATION FROM PAYOR COMPUTER
SECURITY SERVER CREATES TIME STAMP
UPDATES PAYOR PROGRAM TRANSACTION FILES (PAYEE AND PAYOR) AND INCREMENT REGISTER NUMBER
CLOSE CONNECTION
SECURITY SERVER UPDATES PAYOR ACCOUNT

Fig. 4A

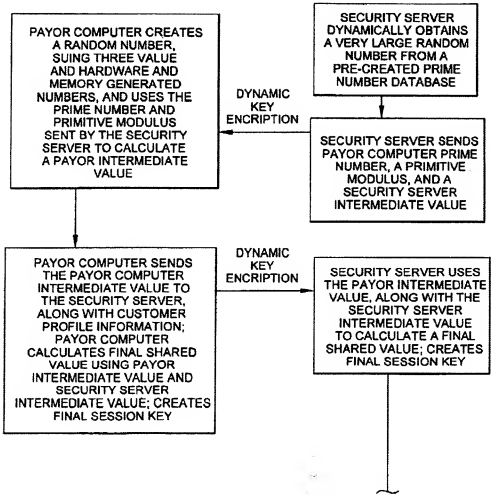
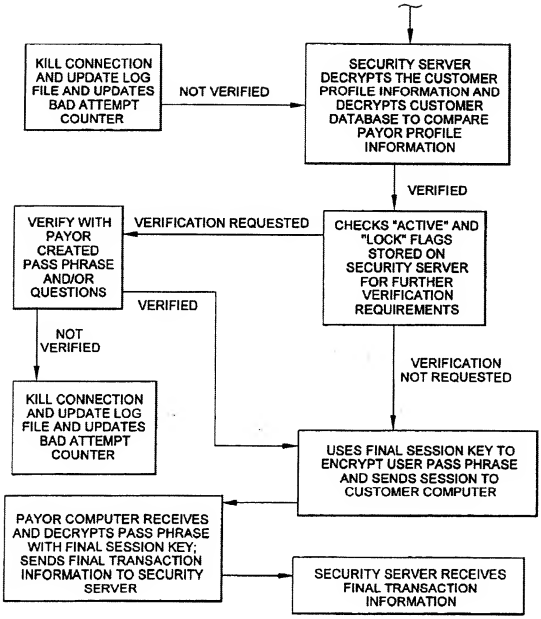

PAYOR COMPUTER CREATES A RANDOM NUMBER, SUING THREE VALUE AND HARDWARE AND MEMORY GENERATED NUMBERS, AND USES THE PRIME NUMBER AND PRIMITIVE MODULUS SENT BY THE SECURITY SERVER TO CALCULATE A PAYOR INTERMEDIATE VALUE
SECURITY SERVER DYNAMICALLY OBTAINS A VERY LARGE RANDOM NUMBER FROM A PRE-CREATED PRIME NUMBER DATABASE
DYNAMIC KEY ENCRIPTION
SECURITY SERVER SENDS PAYOR COMPUTER PRIME NUMBER, A PRIMITIVE MODULUS, AND A SECURITY SERVER INTERMEDIATE VALUE
PAYOR COMPUTER SENDS THE PAYOR COMPUTER INTERMEDIATE VALUE TO THE SECURITY SERVER, ALONG WITH CUSTOMER PROFILE INFORMATION; PAYOR COMPUTER CALCULATES FINAL SHARED VALUE USING PAYOR INTERMEDIATE VALUE AND SECURITY SERVER INTERMEDIATE VALUE; CREATES FINAL SESSION KEY
DYNAMIC KEY ENCRIPTION
SECURITY SERVER USES THE PAYOR INTERMEDIATE VALUE, ALONG WITH THE SECURITY SERVER INTERMEDIATE VALUE TO CALCULATE A FINAL SHARED VALUE; CREATES FINAL SESSION KEY

Fig. 4B


KILL CONNECTION AND UPDATE LOG FILE AND UPDATES BAD ATTEMPT COUNTER
NOT VERIFIED
SECURITY SERVER DECRYPTS THE CUSTOMER PROFILE INFORMATION AND DECRYPTS CUSTOMER DATABASE TO COMPARE PAYOR PROFILE INFORMATION
VERIFIED
VERIFY WITH PAYOR CREATED PASS PHRASE AND/OR QUESTIONS
VERIFICATION REQUESTED
CHECKS "ACTIVE" AND "LOCK" FLAGS STORED ON SECURITY SERVER FOR FURTHER VERIFICATION REQUIREMENTS
VERIFIED
NOT VERIFIED
VERIFICATION NOT REQUESTED
KILL CONNECTION AND UPDATE LOG FILE AND UPDATES BAD ATTEMPT COUNTER
USES FINAL SESSION KEY TO ENCRYPT USER PASS PHRASE AND SENDS SESSION TO CUSTOMER COMPUTER
PAYOR COMPUTER RECEIVES AND DECRYPTS PASS PHRASE WITH FINAL SESSION KEY; SENDS FINAL TRANSACTION INFORMATION TO SECURITY SERVER
SECURITY SERVER RECEIVES FINAL TRANSACTION INFORMATION

SYSTEM, METHOD, AND ARTICLE OF MANUFACTURE FOR SECURE TRANSACTIONS UTILIZING A COMPUTER NETWORK

FIELD OF THE INVENTION

The present invention relates to the secure, electronic payment of consumer debt over a communication network, and more specifically, to a system, method and article of manufacture for securely transmitting payment information from a payor to a security server, which processes the transaction, and returning a confirmation of said payment.

BACKGROUND

The present invention relates to a method, device utilizing an electronic graphical representation of a monetary system for implementing electronic money payments as an alternative medium of economic exchange to cash, checks, credit and debit cards, and traditional electronic funds transfer. The system according to the present invention utilizes electronic representations of money that are designed to be universally accepted and exchanged as economic value by subscribers of the monetary system.

Currently, approximately 350 billion monetary transactions occur between individuals and institutions annually. The extensive use of monetary transactions has limited the automation of individual transactions such as purchases, fares, and bank account deposits and withdrawals. Individual cash transactions are burdened by the need to have the correct amount of cash or providing change therefor. Furthermore, the handling and managing of paper cash and coins is inconvenient, costly, and time consuming for both individuals and financial institutions.

Although checks may be written for any specific amount up to the amount available in the account, checks have very limited transferability and must be supplied from a physical inventory. Paper-based checking systems do not offer sufficient relief from the limitations of cash transactions, sharing many of the inconveniences of handling currency while adding the inherent delays associated with processing checks. To this end, economic exchange is moving toward automation for greater convenience at a lower cost.

Automation is being used for large transactions through computerized electronic funds transfer ("EFT") systems. EFT is essentially a process of value exchange achieved through the banking system's centralized computer transactions. EFT services are a transfer of payments utilizing electronic "letters of credit" and are used primarily by large commercial organizations. The American Clearing House (ACH), where a user can enter a pre-authorized code and download information with billing occurring later, and Point Of Sale (POS) systems, where transactions are processed by connecting with a central computer for authorization for the transaction granted or denied immediately, are examples of EFT systems that are utilized by retail and commercial organizations.

Home banking bill payment services are another example of EFT systems used by individuals to make payments from a home computer. Currently, however, home banking initiatives have found few payors. Less than one percent of bank payors use service accounts for transfers and information, using personal computers over telephone lines. One reason that home banking has not been a successful product is because the payor cannot deposit and withdraw money as needed in this type of system. Another reason home banking

initiatives have found few payors is the inherent distrust in the security of data transmission of financial data across the Internet **100** prevalent in society given the present Internet **100** security and encryption products currently available to the general public.

Current EFT systems, credit cards, or debit cards, which are used in conjunction with an online system to transfer money between accounts, such as between the account of a merchant and that of a payor, do not satisfy the need for an automated transaction system providing an ergonomic interface.

To implement an automated, convenient transaction that can dispense some form of economic value, there has been a trend towards off-line payments. For example, numerous ideas have been proposed for some form of "electronic money" that can be used in non-cash payment transactions as alternatives to the traditional currency and check types of payment systems. Best known of these are magnetic stripe cards purchased for a given amount and from which a prepaid value can be deducted for specific purposes. Upon exhaustion of the economic value, the cards may be thrown away. Other examples include memory cards or so called smart cards, which are capable of repetitively storing information representing value that is likewise deducted for specific purposes. These methods also do not satisfy the current needs for a consumer friendly, convenient and secure electronic transaction system.

The Internet has become a valuable tool for the electronic transfer of information, which can include financial transactions. It is possible and desirable for a computer operating under the control of the payor over a publicly accessible packet-switched network (e.g., the Internet) to bi-directionally share payment information with a computer operated under the control of a payee, without risking the exposure of the information to interception by third parties that have access to the network, and to assure that the information is from an authentic source. It is further desirable for this information, including a subset of the information provided by the payor, to be provided to the payee by the security server system that is designated by a bank or other financial institution that has the responsibility of providing payment on behalf of the payor, without the risk of exposing that information to interception by third parties. Such institutions may include, for example, merchants or financial institutions.

One such attempt to provide such a secure transmission channel is a secure payment technology such as Secure Electronic Transaction (hereinafter "SET"), jointly developed by the Visa and MasterCard card associations, and described in Visa and MasterCard's Secure Electronic Transaction (SET) Specification, Feb. 23, 1996, hereby incorporated by reference. Other such secure payment technologies include Secure Transaction Technology ("STT"), Secure Electronic Payments Protocol ("SEPP"), Internet Keyed Payments ("IKP"), Net Trust, and Cybercash Credit Payment Protocol. One of ordinary skill in the art readily comprehends that any of the secure payment technologies can be substituted for the SET protocol without undue experimentation.

Such secure payment technologies, referenced above, require the payor to operate software that is compliant with the secure payment technology, interacting with third-party certification authorities, thereby allowing the payor to transmit encoded information to a payee, some of which may be decoded by the payee, and some which can be decoded only by an institution specified by the payor.

Another such attempt to provide such a secure transmission channel is a general-purpose secure communication protocol such the Secure Sockets Layer (hereinafter "SSL"). SSL provides a means for secure transmission between two computers. SSL has the advantage that it does not require special-purpose software to be installed on the payor's computer because it is already incorporated into widely available software that many people utilize as their standard Internet access medium. Other examples of general-purpose secure communication protocols include Private Communications Technology ("PCT") from Microsoft, Inc.; Secure Hyper-Text Transport Protocol ("SHTTP") from Terisa Systems; Shen; Kerberos; Photuris; and Pretty Good Privacy ("PGP") all of which meet the IPSEC criteria. One of ordinary skill in the art readily comprehends that any of the general-purpose secure communication protocols can be substituted for the SSL transmission protocol without undue experimentation. However these protocols have proven to be vulnerable to attack, therefore greater security must be available.

Banks desire an Internet payment solution that functions similar to existing Point of Sale (POS) applications that are currently installed on their host computers and require minimal changes to their host systems. This is a critical requirement since any downtime for a bank's host computer system represents an enormous expense. Currently, there are over fourteen hundred different payment-related applications available. The large number of applications is necessary to accommodate a wide variety of host message formats, diverse methods for communicating to a variety of hosts with different dial-up and direct-connect schemes, and different certification around the world.

Internet-based payment solutions require additional security measures that are not found in conventional POS or EFT terminals. This additional requirement is necessitated because Internet communication is done over publicly accessible unsecured communication line in stark contrast to the private, secure, dedicated phone or leased line service utilized between a traditional payee and an acquiring bank. Thus, it is critical that any solution utilizing the Internet for a communication backbone employs some form of secure cryptography.

As discussed above, the current state-of-the-art in Internet based payment processing is a protocol referred to as SET, or Secure Electronic Transaction. Since the SET messages are uniform across all implementations, banks cannot differentiate themselves in any reasonable way. Also, since SET is not a proper superset of all protocols utilized today, there are bank protocols that cannot be mapped or translated into SET because they require data elements for which SET has no placeholder. Further, SET only handles the message types directly related to authorizing and capturing credit card transactions and adjustments to these authorizations or captures. In a typical EFT terminal in the physical world, these messages comprise almost the entire volume of the total number of messages between the payee and the authorizing bank, but only half of the total number of different message types. These message types, which are used infrequently, but which are critical to the operation of the EFT terminal must be supported for proper transaction processing.

Generally, applications written for this field are written using JAVA, C, and/or the C++ languages and utilize object oriented programming methodology. Object oriented programming (OOP) has become increasingly used to develop complex applications. As OOP moves toward the mainstream of software design and development, various software solutions require adaptation to make use of the benefits of OOP.

OOP is a process of developing computer software using objects, including the steps of analyzing the problem, designing the system, and constructing the program. An object is a software package that contains both data and a collection of related structures and procedures. Since it contains both data and a collection of structures and procedures, it can be visualized as a self-sufficient component that does not require other additional structures, procedures or data to perform its specific task. OOP, therefore, views a computer program as a collection of largely autonomous components, called objects, each of which is responsible for a specific task. This concept of packaging data, structures, and procedures together in one component or module is called encapsulation.

In general, OOP components are reusable software modules that present an interface that conforms to an object model and that are accessed at run-time through component integration architecture. Component integration architecture is a set of architecture mechanisms that allow software modules in different process spaces to utilize each other's capabilities or functions. This is generally done by assuming a common component object model on which to build the architecture.

It is worthwhile to differentiate between an object and a class of objects at this point. An object is a single instance of the class of objects, which is often just called a class. A class of objects can be viewed as a blueprint, from which many objects can be formed.

OOP allows the programmer to create an object that is a part of another object. For example, the object representing a piston engine is said to have a composition-relationship with the object representing a piston. In reality, a piston engine comprises a piston, valves and many other components; the fact that a piston is an element of a piston engine can be logically and semantically represented in OOP by two objects.

OOP also allows creation of an object that "depends on" another object. If there are two objects, one representing a piston engine and the other representing a piston engine wherein the piston is made of ceramic, then the relationship between the two objects is not that of composition. A ceramic piston engine does not make up a piston engine. Rather it is merely one kind of piston engine that has one more limitation than the piston engine; its piston is made of ceramic. In this case, the object representing the ceramic piston engine is called a derived object, and it inherits all of the aspects of the object representing the piston engine and adds further limitation or detail to it. The object representing the ceramic piston engine "depends from" the object representing the piston engine. The relationship between these objects is called inheritance.

When the object or class representing the ceramic piston engine inherits all of the aspects of the objects representing the piston engine, it inherits the thermal characteristics of a standard piston defined in the piston engine class. However, the ceramic piston engine object overrides these ceramic specific thermal characteristics, which are typically different from those associated with a metal piston. It skips over the original and uses new functions related to ceramic pistons. Different kinds of piston engines have different characteristics, but may have the same underlying functions associated with it (e.g., how many pistons in the engine, ignition sequences, lubrication, etc.). To access each of these functions in any piston engine object, a programmer would call the same functions with the same names, but each type of piston engine may have different/overriding implemen-

tations of functions behind the same name. This ability to hide different implementations of a function behind the same name is called polymorphism and it greatly simplifies communication among objects.

With the concepts of composition-relationship, encapsulation, inheritance and polymorphism, an object can represent just about anything in the real world. In fact, our logical perception of the reality is the only limit on determining the kinds of things that can become objects in object-oriented software. Some typical categories are illustrated as follows: objects can represent physical objects, such as automobiles in a traffic-flow simulation, electrical components in a circuit-design program, financial transactions in an economics model, or aircraft in an air-traffic-control system; objects can represent elements of the computer-user environment such as windows, menus or graphics objects; an object can represent an inventory, such as a personnel file or a table of the latitudes and longitudes of cities; or an object can represent user-defined data types such as time, angles, and complex numbers, or points on the plane.

Programming languages are beginning to fully support the OOP principles, such as encapsulation, inheritance, polymorphism, and composition-relationship. With the advent of the C++ language, many commercial software developers have embraced OOP. C++ is an OOP language that offers a fast, machine-executable code. Furthermore, C++ is suitable for both commercial-application and systems-programming projects. For now, C++ appears to be the most popular choice among many OOP programmers, but there is a host of other OOP languages, such as Smalltalk, common lisp object system (CLOS), and Eiffel. Additionally, OOP capabilities are being added to more traditional popular computer programming languages such as Pascal.

The development of graphical user interfaces began to turn procedural programming arrangements inside out. These interfaces allow the user, rather than program logic, to drive the program and decide when certain actions should be performed. Today, most personal computer software accomplishes this by means of an event loop that monitors the mouse, keyboard, and other sources of external events and calls the appropriate parts of the programmer's code according to actions that the user performs. The programmer no longer determines the order in which events occur. Instead, a program is divided into separate pieces that are called at unpredictable times and in an unpredictable order. By relinquishing control in this way to users, the developer creates a program that is much easier to use. Nevertheless, individual pieces of the program written by the developer still call libraries or objects provided by the operating system to accomplish certain tasks, and the programmer must still determine the flow of control within each piece after it's called by the event loop. Application code still "sits on top of" the system.

Even event loop programs require programmers to write a lot of code that should not need to be written separately for every application. The concept of an application framework carries the event loop concept further. Instead of dealing with all the nuts and bolts of constructing basic menus, windows, and dialog boxes and then making these things all work together, programmers using application frameworks start with working application code and basic user interface elements in place. Subsequently, they build from there by replacing some of the generic capabilities of the framework with the specific capabilities of the intended application.

Application frameworks reduce the total amount of code that a programmer has to write from scratch. However, because the framework is really a generic application that displays windows, supports copy and paste, and so on, the programmer can also relinquish control to a greater degree than event loop programs permit. The framework code takes care of almost all event handling and flow of control, and the programmer's code is called only when the framework needs it (e.g., to create or manipulate a proprietary data structure).

A programmer writing a framework program not only relinquishes control to the user (as is also true for event loop programs), but also relinquishes the detailed flow of control within the program to the framework. This approach allows the creation of more complex systems that work together in interesting ways, as opposed to isolated programs, having custom code, being created over and over again for similar problems.

Thus, as is explained above, a framework basically is a collection of cooperating classes of objects that make up a reusable design solution for a given problem domain. It typically includes objects that provide default behavior (e.g., for menus and windows), and programmers use it by inheriting some of that default behavior and overriding other behavior so that the framework calls application code at the appropriate times. There are three main differences between frameworks and class libraries:

Behavior versus Protocol. Class libraries are essentially collections of behaviors that you can call when you want those individual behaviors in your program. A framework, on the other hand, provides not only behavior but also the protocol or set of rules that govern the ways in which behaviors can be combined, including rules for what a programmer is supposed to provide versus what the framework provides.

Call versus Override. With a class library, the programmer codes objects and calls their member functions. It's possible to code and call objects in the same way with a framework (i.e., s to treat the framework as a class library), but to take full advantage of a framework's reusable design, a programmer typically writes code that overrides and is called by the framework. The framework manages the flow of control among its objects. Writing a program involves dividing responsibilities among the various pieces of software that are called by the framework rather than specifying how the different pieces should work together.

Implementation versus Design. With class libraries, programmers reuse only implementations, whereas with frameworks, they reuse design. A framework embodies the way a family of related programs or pieces of software work. It represents a generic design solution that can be adapted to a variety of specific problems in a given domain. For example, a single framework can embody the way a user interface works, even though two different user interfaces created with the same framework might solve quite different interface problems.

Thus, through the development of frameworks for solutions to various problems and programming tasks, significant reductions in the design and development effort for software can be achieved.

To date, Web development tools have been limited in their ability to create dynamic Web applications that span from client to server and inter-operate with existing computing resources. Until recently, HTML has been the dominant technology used in development of Web-based solutions. However, HTML has proven to be inadequate in the following areas: poor performance; restricted user interface capabilities; lack of interoperability with existing applications and data; inability to scale, and weak security.

Sun Microsystem's Java language solves many problems by: improving performance; enabling the creation of dynamic, real-time web applications; and providing the ability to create a wide variety of user interface components.

With Java, developers can create robust User Interface (UI) components. Custom "widgets" (e.g. real-time stock tickers, animated icons, etc.) can be created, and performance is improved. Unlike HTML, Java supports the notion of validation, offloading appropriate processing onto the client for improved performance. Dynamic, real-time Web pages can be created. Using the above-mentioned custom UI components, dynamic Web pages can also be created.

Sun's Java language has emerged as an industry-recognized language for "programming the Internet." Sun defines Java as: "A simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded, dynamic, buzzword-compliant, general-purpose programming language. Java supports programming for the Internet in the form of platform-independent Java applets." Java applets are small, specialized applications that comply with Sun's Java Application Programming Interface (API) allowing developers to add "interactive content" to Web documents (e.g. simple animations, page adornments, basic games, etc.). Applets execute within a Java-compatible browser (e.g. Netscape Navigator or Internet Explorer) by copying code from the server to client. From a language standpoint, Java's core feature set is based on C++. Sun's Java literature states that Java is basically "C++, with extensions from Objective C for more dynamic method resolution".

Another technology that provides similar function to JAVA is provided by Microsoft and ActiveX Technologies, to give developers and Web designers the wherewithal to build dynamic content for the Internet and personal computers. ActiveX includes tools for developing animation, 3-D virtual reality, video and other multimedia content. The tools use Internet standards, work on multiple platforms, and are being supported by over 100 companies. The group's building blocks are called ActiveX Controls, small, fast components that enable developers to embed parts of software in hypertext markup language (HTML) pages. ActiveX Controls work with a variety of programming languages including Microsoft Visual C++, Borland Delphi, Microsoft Visual Basic programming system and Microsoft's development tool **10** for Java, code named "Jakarta." ActiveX Technologies also includes ActiveX Server Framework, allowing developers to create server applications. One of ordinary skill in the art readily recognizes that ActiveX could be substituted for JAVA without undue experimentation to practice the invention.

SUMMARY OF THE INVENTION

According to a broad aspect of a preferred embodiment of the invention, secure transmission of data is provided between at least two computer systems over a public communication system, such as the Internet. Secure transmission of data is provided from the payor computer system to a banking computer system, which may initiate further secure transmission of payment information regarding a payment instrument from the banking computer system to a the payee computer system. The payment system formats transaction information appropriately and transmits the transaction to the particular host system. The host system evaluates the payment information and returns a level of authorization of credit transfer to the payee computer.

The novel features that are considered characteristic of the invention are set forth with particularity in the appended claims. The invention itself, however, both as to its structure and its operation together with the additional object and advantages thereof will best be understood from the following description of the preferred embodiment of the present invention when read in conjunction with the accompanying drawings. Unless specifically noted, it is intended that the words and phrases in the specification and claims be given the ordinary and accustomed meaning to those of ordinary skill in the applicable art or arts. If any other meaning is intended, the specification will specifically state that a special meaning is being applied to a word or phrase. Likewise, the use of the words "function" or "means" in the Description of Preferred Embodiments is not intended to indicate a desire to invoke the special provision of 35 U.S.C. §112, paragraph 6 to define the invention. To the contrary, if the provisions of 35 U.S.C. §112, paragraph 6, are sought to be invoked to define the invention(s), the claims will specifically state the phrases "means for" or "step for" and a function, without also reciting in such phrases any structure, material, or act in support of the function. Even when the claims recite a "means for" or "step for" performing a function, if they also recite any structure, material or acts in support of that means of step, then the intention is not to invoke the provisions of 35 U.S.C. §112, paragraph 6. Moreover, even if the provisions of 35 U.S.C. §112, paragraph 6, are invoked to define the inventions, it is intended that the inventions not be limited only to the specific structure, material or acts that are described in the preferred embodiments, but in addition, include any and all structures, materials or acts that perform the claimed function, along with any and all known or later-developed equivalent structures, materials or acts for performing the claimed function.

DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages are better understood from the following detailed description of a preferred embodiment of the invention with reference to the drawings, in which:

FIG. **1** is a block diagram of a representative hardware environment in accordance with a preferred embodiment;

FIG. **2** depicts an overview in accordance with a preferred embodiment;

FIG. **3** is a block diagram of the system in accordance with a preferred embodiment;

FIG. **4** depicts a preferred embodiment of an acceptance method according to the present invention.

DETAILED DESCRIPTION

The present invention is a system or method and device useful for the secure electronic payment of consumer debts over a publicly accessible computer network.

A preferred embodiment of a system in accordance with the present invention is practiced in the context of personal computers or workstations. A representative hardware environment is depicted in FIG. **1**, which illustrates a typical hardware configuration of a computer workstation in accordance with a preferred embodiment having a central processing unit **10**, such as a microprocessor, and a number of other units interconnected via a system bus **12**. The workstation shown in FIG. **1** includes Random Access Memory (RAM) **14**, Read Only Memory (ROM) **16**, an I/O adapter **18** for connecting peripheral devices, such as disk storage units **20** to the bus **12**, a user interface adapter **22** for connecting a keyboard **24**, a mouse **26**, a speaker **28**, a

microphone **32**, and/or other user interface devices, such as a touch screen or and the like (not shown) to the bus **12**, communication adapter **34** for connecting the workstation to a communication network **100** (e.g., a data processing network) and a display adapter **36** for connecting the bus **12** to a display device **38**. The workstation typically has resident thereon an operating system such as the Microsoft Windows NT or Windows/95 Operating System (OS), the IBM OS/2 operating system, the MAC OS, or UNIX operating system. Those skilled in the art will appreciate that the present invention may also be implemented on platforms and operating systems other than those mentioned.

The preferred embodiment of the invention utilizes a variety of different software languages, (preferably CC++ and JAVA but may include such languages as HyperText Markup Language (HTML) and Extended Markup Language (XML)), to implement objects and documents on the Internet **100** together with a general-purpose secure communication protocol for a transport medium between the client and the payee.

FIG. **3** depicts an overview of the present invention. The preferred form of the present invention uses at least two separate, but compatible, software packages. Security server software that continuously runs on a security server **140** and payor software that runs on demand on a payor computer system **120**. The payor computer system **120** communicates via the payor software with the security server **140** via the security server software. The communication, or transaction, session operates under the secure communication protocol described below. A payee computer system **130** may also communicate via payee software with the security server **140**.

A security server **140** is a computer system that provides electronic commerce services in support of a bank or other financial institution, and that interfaces to the financial institution to support the authorization and capture of transactions. The transaction session between the payor computer system **120** and the security server **140** operates under a variant of a secure payment technology, as described herein, referred to as Payor-Originated Secure Electronic Transactions ("POSET"), as is more fully described herein.

Initially, the payor creates a one-time payor profile at the receiving institution, such as a bank. The payor profile preferably includes a user pass phrase and user created personal verification questions, which are used for future verification of payor identity. The verification questions are randomly created by the payor and may be questions such as mother's maiden name, favorite color or the like. Preferably the pass phrase is not limited to a short contiguous number and/or letter combination, like an ordinary pin or password, but can include blank or white spaces between characters or words. The use of a longer phrase helps to prevent a "dictionary attack" on the pass phrase. The benefit to the use of a phrase with white spaces is the increased ease with which the payor can remember a more complicated pass phrase, thereby increasing security. The payor profile information is encrypted and made resident on the security server **140**. The encryption of the payor profile information adds a level of security against unauthorized access by institution or bank personnel.

Generally, a security server program is resident and continuously running on the security server **140**. This allows the security server program to be accessed at any time by a payor. The payor launches the payor software program on the payor computer system **120**. Upon the first launch of the payor program, the payor program executes an initialization, or registration procedure. In this procedure, the payor computer program requires the payor to create and enter a customer, or access, password. This password is used on each subsequent launch of the payor program to verify and identify the payor and account or accounts being accessed. A user transaction screen, such as a "check screen" is then presented to the payor. In the transaction screen are transaction fields that the payor fills in and sends to the security server program. Once a transaction has been initialized, the payor computer program then attempts to contact the security server **140** over a computer network system, commonly known as the Internet **100**.

During the first session, after communications have been established between the security server **140** and the payor computer system **120**, in one embodiment the security server program obtains hardware specific information from the payor computer system **120**. Since hardware specific information is individual to each computer, it acts as a "finger print" that can be used to uniquely identify the computer program. Additionally, this allows easy verification and identification of the specific computer program during future transactions. The hardware specific information is encrypted and stored with the payor profile information in the security server database or on both computers to account for subsequent changes in the hardware configuration, i.e., new hard drives and the like. By combining use of the customer password, the payor computer hardware information, and payor profile information, both the payor and the payor's program may be quickly, easily, and securely verified during future transactions.

When initiating communication with the security server system **140**, the payor computer system **120** may use any well-known access protocol, e.g., Transmission Control Protocol/Internet Protocol ("TCP/IP"). A description of TCP/IP is provided in Information Sciences Institute, "Transmission Control Protocol DARPA Internet Program Protocol Specification (RFC 793)" (September 1981), and Information Sciences Institute, "Internet Protocol DARPA Internet Program Protocol Specification (RFC 791)" (September 1981). In this implementation, the payor computer system **120** acts as a client and the security server system **140** acts as a server.

When initiating communication with the security server **140**, the payor computer system **120** first sends a "client request for connection" message to the security server system **140**. The client request for connection message may further include a variable length session identifier.

In response to the client request for connection message, if the security server system **140** wishes to correspond with the payor computer system **120**, it responds with a message to the payor computer system **120** to switch to a second, separate transaction port, thereby creating a second, or transaction, session. An alternate way to consider this is as a single session that is conducted over two separate pathways: 1) over a first initial communications port; and 2) over a second transaction port. This is an important part of the present invention in that the identity of the second port is dynamically assigned and changes for each and every independently created session. This creates an extra element of variability to the transaction process that improves to the security of the transaction. If the security server system **140** does not wish to communicate with the payor computer system **120**, it responds with a message indicating refusal to communicate.

FIG. **4** depicts the detailed steps of authorizing communications between the payor computer **120** and the security

server **140**, including the generating and transmitting of a payment authorization request.

Preferably, the security server connection message includes an initial very large prime number, a prime modulus, and a server calculated intermediate value. The payor computer system **120** replies to the security server connection message with a client response message that preferably includes a payor calculated intermediate value. Separately, the payor computer system **120** calculates a final shared value. Once the security server **140** receives the payor calculated intermediate value, it too calculates the final shared value.

More specifically, the security server system **140** obtains a randomly generated server secret number. The security server **140** also selects a very large public prime number, which is preferably, a very large prime number residing in a pre-created prime number database, and a prime modulus. The security server **140** creates a server calculated intermediate value using the secret random number, the public prime number, and the prime modulus, by performing a portion of a selected algorithm. The security server **140** sends the server calculated intermediate value, the public prime number, and the prime modulus to the payor computer system **120**. The payor computer system **120** generates a payor secret random number and uses the public prime number and prime modulus to create a payor calculated intermediate value. Additionally, the payor computer system **120** uses the server calculated intermediate value and the payor calculated intermediate value to calculate a shared final value. The payor computer system **120** sends the payor calculated intermediate value, along with selected payor ID or profile information to the security server **140**. In a preferred embodiment, the calculated intermediate values are encrypted before transmission. The security server **140** uses the payor calculated intermediate value, with the security server calculated intermediate value, to also calculate the same shared final value. Thus, the shared final value, which while known by both computers, is never transmitted. For this very reason, the shared final value cannot be intercepted by a third party for use in a fraudulent attempt on the account.

While other like algorithms with similar properties may be used, a preferred algorithm for the above security process is as follows:

SIV (security server intermediate value)=g^{SRN} mod p,

where g is the public prime number, mod p is the prime modulus, and SRN is the security server random number,

CIV (payor intermediate value)=g^{CRN} mod p,

where g is the public prime number, mod p is the prime modulus, and CRN is the payor random number; and

SFV (shared final value)=SIV^{CRN} mod p, for the payor computer **120**, and

=CIV^{SRN} mod p, for the security server **140**.

In creating the payor random number, a preferred embodiment has the payor computer system **120** using random values to seed the creation of a unique large random number. Preferably, these random numbers are obtained dynamically within the session, and even more preferably, are obtained from unique non-repeatable functions, such as mouse or cursor positions.

In each separate system, the payor computer system **120** and the security server **140**, the shared final value, in combination with selected portions of the payor profile information, is encrypted using yet another function, such as a one-way secure hash algorithm, to produce a final session key. This final session key, having portions that are generated dynamically within each session, and portions that are personal to each individual payor, is computationally impossible to decode or generate in the time allotted for each transaction, thereby providing an exceptionally high level of security for each transaction. The inclusion of a one-way function encryption of the data provides an ultra-high level of security for each transaction.

The security server **140** then takes the payor's ID information and compares to the payor profile information resident in the security server **140** in order to verify the payor. If the payor ID information is not verified, the connection is immediately terminated and logged to a failure database.

If the payor profile information is verified during the transaction, the security server **140** proceeds to check several flags, or indicia fields, for an indication of whether further verification or encryption, such as by providing the answers to the private questions previously supplied, is required.

At this point both the payor computer system **120** and the security server system **140** have: 1) negotiated a communication session; 2) have communicated to each other the basis for the calculation of a set of encryption keys that may be used to encrypt and decrypt further communications between the two computer systems, **120** and **140** respectively; and 3) have calculated a final session key that is never transmitted and will be used for further encryption. The payor computer system **120** and the security server system **140** may thereafter engage in a secure financial transaction with a greatly reduced risk of interception or fraud by third parties.

After a connection has been authorized and implemented, the security server **140** checks the present account register number of the payor program. Initially, the current account register number is set to a zero transaction number. At any time, including immediately after registration of the program but before a first transaction has been processed, the payor may simply elect to exit the payor computer program and reenter it at another time. If the payor does not exit the payor program, a graphical user program interface, as discussed above, preferably the "check" screen, is generated on the payor computer **120**. The program interface includes enterable fields for transaction specific information.

If additional encryption is requested, the security server **140** may request portions of transaction information previously sent or it may encrypt a verifying value. If the verifying value is sent to the payor computer system **120**, the payor computer system **120** decrypts the verifying value and uses it in one of two different ways. First, it may be used as additional data added to the transaction information, re-encrypted and sent to the security server. Then, when the security server **140** decrypts the transaction it compares the verifying value before processing the transaction. Second, the decrypted verifying value may be used as an initial value to roll-over encrypt the transaction information, which is then further encrypted using the final session key and sent to the security server **140**. The security server **140** then decrypts the message, and decrypts the roll-over encrypted transaction (using the verifying value). If the both systems use the same verifying value, the security server then has transaction information that is appropriate for the system, otherwise the decryption of the rolled-over information will yield strange characters and/or information. These two methods are typically selected by the software and may be dynamically chosen such that any individual transaction may use one or the other method.

Once the transaction has been verified and processed, the security server **140** creates a time stamp, encrypts it, and

sends it to the payor computer system **120** to finalize the transaction. In this way, the payor, not being in control of the time stamp, cannot create a false time record. Once the payor program receives the time stamp, it then increments the account register number counter of the payor program by one and fills in the check information.

Among the information communicated by the payor computer system **120** to the security server system **140** may be information that specify payment information, such as payee identification, bank identification, bank account numbers, credit card numbers, and related information, collectively referred to as "payment information," that may be used to pay the bill for the goods and/or services ordered. In order to obtain payment, the payee may supply a portion of this information to the bank or other institution responsible for the proffered payment method. This enables the payee to perform payment authorization and payment capture. Payment authorization is the process by which permission is granted to a security server **140** operating on behalf of a financial institution to authorize payment on behalf of the financial institution. This is a process that assesses transaction risk, confirms that a given transaction does not raise the account holder's debt above the account's balance. Payment capture is the process that triggers the movement of funds from the financial institution to the payee's account in order to settle the account.

The security server system **140** identifies the payee for which the transaction is authorized by inspection of the transaction information. The security server system **140** may contact the appropriate payee using a secure means, preferably via the Internet, and using prior art means, obtains a response indicating whether the requested payment is due, presented, and has been confirmed.

In contacting the payee, the security server may utilize one of two different methods. A first method is used for non-institutional payee's, such as private individuals or small businesses. In this method, the security server program automatically generates an electronic mail message (e-mail) that identifies the payor and the fact that a payment has been made. It is preferable that the e-mail message does not indicate the amount of payment or account to which the payment was made for security purposes. A second method, which is preferably used for larger payee's such as large business and institutions is the use of a payee program on a payee computer system **130**. The payee program communicates with the security server program, as detailed below, and may provide, among other information, the name of the payor, the invoice number or customer number, the amount of payment, the account to which the payment has been made, and the like. The transaction between the payee computer program and the security server computer program may be accomplished in either a batch mode or in a continuous, real-time action.

Upon the first launch of the payee program, the payee program executes an initialization, or registration procedure. In this procedure, the payee computer program requires the payee to create and enter a payee, or access, password. This password is used on each subsequent launch of the payee program to verify and identify the payee and account or accounts being accessed. The payee computer system **130** then contacts the security server **140** over a computer network system, commonly known as the Internet **100**. The payee program communicates with the security server program and registers the payee program with the security server program. This registration confirms the identity of the payee computer program.

During a first transaction session, after communications have been established between the security server **140** and the payee computer system **130**, the security server **140** preferably obtains hardware specific information from the payee computer system **130** and stores it in both places to account for changes in the hardware configuration of the payee computer **130**. Since hardware specific information is individual to each computer, it acts as a "finger print" that can be used to uniquely identify the computer. Additionally, this allows easy verification and identification of the specific computer during future transactions. The hardware specific information is encrypted and stored with the payee ID information in the security server database. By combining use of the payee password, the payee computer hardware information, and payee profile information, both the payee and the payee's computer program may be quickly, easily, and securely verified during future transactions.

When initiating communication with the security server system **140**, the payee computer system **130** may use any well-known access protocol, e.g., Transmission Control Protocol/Internet Protocol ("TCP/IP"). A description of TCP/IP is provided in Information Sciences Institute, "Transmission Control Protocol DARPA Internet Program Protocol Specification (RFC 793)" (September 1981), and Information Sciences Institute, "Internet Protocol DARPA Internet Program Protocol Specification (RFC 791)" (September 1981). In this implementation, the payee computer system **130** acts as a client and the security server system **140** acts as a server. It should be noted that the communication may be initiated by the security server program to the payee program with the security server system **140** acting as the client and the payee computer system **130** acting as the server.

When initiating communication with the security server **140**, the payee computer system **130** first sends a "payee request for connection" message to the security server system **140**. The payee request for connection message may further include a variable length session identifier.

In response to the payee request for connection message, if the security server system **140** wishes to correspond with the payee computer system **130**, it responds with a message to the payee computer system **130** to switch to a second, separate transaction port, thereby creating a second, or transaction, session. Another way of thinking about this is as a single session with two separate pathways: 1) a first port for initializing communications; and 2) a second port for transmission of transaction information. This is an important part of the present invention in that the identity of the second port is dynamically assigned and changes for each and every independently created session. This creates an extra element of variability to the transaction process that improves to the security of the transaction. If the security server system **140** does not wish to communicate with the payee computer system **130**, it responds with a message indicating refusal to communicate.

FIG. **4** depicts the detailed steps of authorizing communications between the payee computer **130** and the security server **140**, including the generating and transmitting of a payment authorization request.

Preferably, the security server connection message includes an initial very large prime number, a prime modulus, and a server calculated intermediate value. The payee computer system **130** replies to the security server connection message with a payee response message that preferably includes a payee calculated intermediate value. Separately, the payee computer system **120** calculates a final shared value. Once the security server **140** receives the payee calculated intermediate value, it too calculates the final shared value.

More specifically, the security server system **140** obtains a randomly generated server secret number. The security server **140** also selects a public prime number, which is preferably, a very large prime number residing in a pre-created prime number database, and a prime modulus. The security server **140** creates a server calculated intermediate value using the secret random number, the public prime number, and the prime modulus, by performing a portion of a selected algorithm. The security server **140** sends the server calculated intermediate value, the public prime number, and the prime modulus to the payee computer system **130**. The payee computer system **130** generates a payee secret random number and uses the public prime number and prime modulus to create a payee calculated intermediate value. Additionally, the payee computer system **130** uses the server calculated intermediate value and the payee calculated intermediate value to calculate a shared final value. The payee computer system **130** sends the payee calculated intermediate value, along with selected payee ID information to the security server **140**. In a preferred embodiment, the calculated intermediate values are encrypted before transmission. The security server **140** uses the payee calculated intermediate value, with the security server calculated intermediate value, to also calculate the same shared final value. Thus, the shared final value, which while known by both computers, is never transmitted. For this very reason, the shared final value cannot be intercepted by a third party for use in a fraudulent attempt on the account.

While other like algorithms with similar properties may be used, a preferred algorithm for the above security process is as follows:

SIV (security server intermediate value)=g^{SRN} mod p,

where g is the public prime number, mod p is the prime modulus, and SRN is the security server random number;

MIV (payee intermediate value)=g^{MRN} mod p,

where g is the public prime number, mod p is the prime modulus, and MRN is the payee random number; and

SFV (shared final value)=SIV^{MRN} mod p, for the payee computer **130**, and

=MIV^{SRN} mod p, for the security server **140**.

In creating the payee random number, a preferred embodiment has the payee computer system **130** using random values to seed the creation of a unique large random number. Preferably, these random numbers are obtained dynamically within the session, and even more preferably, are obtained from unique non-repeatable functions, such as mouse or cursor positions, line voltages, or the like.

In each separate system, the payee computer system **130** and the security server **140**, the shared final value, in combination with selected portions of the payee profile information, is encrypted using yet another function, such as a one-way secure hash algorithm, to produce a final session key. This final session key, having portions that are generated dynamically within each session, and portions that are personal to each individual payee , is computationally impossible to decode or generate in the time allotted for each transaction, thereby providing an exceptionally high level of security for each transaction. The inclusion of a one-way function encryption of the data provides an ultra-high level of security for each transaction.

The security server **140** then takes the payee's profile information and compares to the payee profile information resident in the security server **140** in order to verify the payee. If the payee profile information is not verified, the connection is immediately terminated and logged to a failure database.

If the payee profile-information is verified during the transaction, the security server **140** proceeds to check several flags, or indicia fields, for an indication of whether further verification or encryption is required.

At this point both the payee computer system **130** and the security server system **140** have: 1) negotiated a communication session; 2) have communicated to each other the basis for the calculation of a set of encryption keys that may be used to encrypt and decrypt further communications between the two computer systems, **130** and **140** respectively; and **3**) have calculated a final session key that is never transmitted and will be used for further encryption. The payee computer system **130** and the security server system **140** may thereafter engage in a secure financial transaction with a greatly reduced risk of interception or fraud by third parties.

If additional encryption is requested, the security server **140** uses the final session key and encrypts a verifying value. The verifying value is sent to the payee computer system **130**. The payee computer system **130** decrypts the verifying value. The decrypted verifying value is then used in one of two different ways. First, it may be used as additional data added to the transaction information, re-encrypted and sent to the security server. Then, when the security server **140** decrypts the transaction it compares the verifying value before processing the transaction. Second, the decrypted verifying value may be used as an initial value to roll-over encrypt the transaction information, which is then further encrypted using the final session key and sent to the security server **140**. The security server **140** then decrypts the message, and decrypts the roll-over encrypted transaction (using the verifying value). If the both systems use the same verifying value, the security server then has transaction information that is appropriate for the system, otherwise the decryption of the rolled-over information will yield strange characters and/or information. These two methods are typically selected by the software and may be dynamically chosen such that any individual transaction may use one or the other method.

Once the transaction has been verified and processed, the security server **140** creates a time stamp and sends it to the payee computer system **130** to finalize the transaction. In this way, the payee, not being in control of the time stamp, cannot create a false time record.

For the above payee-security server transaction, the payee computer system **130** generates a payee payment capture request and transmits it to the security server system **140**. The security server **140** processes the payment capture request, generates a payment capture response and transmits it to the payee computer system **130**. The payee computer system **130** processes payment capture response and verifies that payment for the goods or services purchased by the payor have been captured. The basic capture request is a data area that includes all the information needed by the security server system **140** to trigger a transfer of funds to the payee operating the payee computer system **130**.

Specifically, a capture request includes, as a minimum amount of information, a capture request amount, a date, and a Payee ID (MID) for the particular payee.

The security server system **140** creates a basic capture response. The basic capture response is a data area that includes all the information to indicate whether a capture request was granted or denied.

A Virtual Point of Payment (vPOP) software is also described in accordance with a preferred embodiment using smart card technology or kiosk technology. The vPOP software provides payment functionality on independent

platforms, allowing a payor to process payments securely using a smart card and the Internet **100**. It provides full payment functionality for a variety of payment instruments.

A brief description of the vPOP software functions are provided below. The vPOP provides an interface for transactions that are initiated by the consumer. The consumer initiates a transaction from a Graphical User Interface (GUI) and all the transactions that are initiated by the consumer using a smart card and are routed through a remote computer or kiosk to the security server.

The payment functionality provided by the vPOP software is "Consumer-Initiated" at a site remote from the payee computer system **130**. The normal flow of a transaction is via the vPOP software into a security server software that is responsible for converting into the appropriate format for additional processing and forwarding to existing host payment authorization systems.

Smart cards, according to the present invention has a cyclic registry that is used for transaction data storage. There are at least two separate registers in which at least two separate transactions may be stored. The actual number of registers is only limited by the available space in the memory of the smart card hardware. Additionally, each smart card must be registered to each individual at the financial institution, like a credit card, to prevent unauthorized access. This includes the use of a pin number or pass phrase to access the functionality of the smart card. Finally, the smart card may have encrypted verification information, such as portions of the above described payee profile information, which is used by the security server to securely identify the payor.

In use, the smart card is inserted into a kiosk computer having a modified version of the payee computer program running (the modification being the lack of a registry memory function). The payor is required to provide the smart card pin number or pass phrase. Once the payor correctly provides the smart card pin number or pass phrase, the transaction (check) screen is presented with transaction header information, which is encrypted and stored on the smart card, already filled in. The payor fills in the applicable fields and sends the transaction to the security server program using the same encryption and verification process as described above.

If the payor repeatedly provides an incorrect pin number or pass phrase, or if the security server program has the smart card flagged as missing or stolen, then the security server program sends a message to the kiosk computer to keep the smart card and not release it to the user. Alternately, the security server may send a message to the kiosk computer deactivating the smart card at the kiosk computer. In yet another embodiment, the security server periodically uploads to the kiosk computer a list of missing or stolen smart cards. In this embodiment, the kiosk computer reads the identification of the smart card upon insertion and, upon identification of the smart card as flagged, refuses to allow access to the kiosk program or transaction processing and may or may not keep the smart card. Additionally, the security server may communicate the time and location of the use of the stolen smart card to the proper authorities.

Host Payment Functionality: these transactions require communication with the security server **140**, either immediately or at a later stage. For example, an Online Authorization-Only transaction, when initiated, communicates with the host immediately. However, an Off-line Authorization-Only transaction is locally authorized by the vPOP software without having to communicate with the host, but at a later stage this off-line authorization transaction is sent to the host. Within the Host Payment Functionality some transactions have an associated Payment Instrument, while others do not. These two kinds of transactions are:

Host Financial Payment Functionality: these transactions have a Payment Instrument (Smart card, Credit Card, Debit Card, E-Cash, E-Check, etc.) associated with them.

Host Administrative Payment Functionality: these transactions do not require a payment instrument, and provide either administrative or inquiry functionality. Examples of these transactions are "Reconcile" or the "Batch Close."

Local Functions and Transactions: these transactions do not require communication with the host at any stage, and provide essential vPOP software administrative functionality. An example of this is the vPOP software configuration function, which is required to set up the vPOP software. Another example is the "vPOP Batch Review" function, which is required to review the different transactions in the vPOP Batch or the Transaction Log.

A preferred embodiment of the vPOP software supports various Payment Instruments. A consumer chooses a payment based on personal preferences. Some of the Payment Instruments supported include credit cards, debit cards, electronic cash, electronic checks, and micro-payments (electronic coin).

As discussed above, the different Payment Functionality provided by the vPOP terminal require communication with the security server **140** and these transactions are referred to as "Online Transactions." The transactions that can be done locally without having to communicate are referred to as "Local Functions/Transactions." In order to provide support for many different types of Payment Instruments, the vPOP Payment Functionality have been categorized.

An authorization without capture transaction is used to validate the card holder's account number for a payment that needs to be performed at a later stage. The transaction does not confirm a payment's completion to the host, and there is no host data capture in this event. The vPOP captures this transaction record and later forwards it to the host to confirm the payment in a forced post transaction request.

A forced post transaction confirms to a host computer that a completion of a payment has been accomplished and requests data capture of the transaction. The forced post transaction is used as a follow-up transaction after doing an authorization (Online or Off-line) transaction.

The offline post transaction is identical to the "authorization without capture" transaction, except that the transaction is locally captured by the vPOP without initiating communication with a host. A forced post operation is done as a follow-up operation of this transaction.

The Internet **100** provides the communication processing necessary to enable the vPOP software. The software interface CGI communicates via the Internet **100** to provide information to the vPOP Security Server **140**, which formats information in accordance with the vPOP.

As discussed above, in order to actually transact business over the Internet **100**, the user must first register the smart card with the bank with which he signs an acquiring agreement. For online payment processing, the user must also create an appropriate set of digital credentials in the form of personal questions and possibly additional passwords, depending on the financial institution and/or user's desires.

The user, interacting with the software, can navigate to a list of security servers, and selects the bank to connect to by selecting from the list of banks.

Each vPOP may process a single transaction at a time. Security Servers **140** can process many transactions at a

time, so transaction requests can often occur simultaneously at the security server **140**. Thus, the security server **140** version of the vPOP Software must have support for multi-tasking and provide support for multiple threads to be active at the same time in the same system as well as the same process. This requirement is relatively straightforward.

Since the Internet **100** is so pervasive, and access is available from virtually any computer, utilizing the Internet **100** as the communication backbone for connecting the payor, payee and access to the authorizing bank through a security server **140** allows the payee vPOP software to be remotely located from the payee's premises. For example, the payor could pay for goods from any computer system attached to the Internet **100** at any location in the world. Similarly, the payee vPOP system could be located at a central host site where payee vPOP systems for various payees all resided on a single host with their separate access points to the Internet **100**. The payee could utilize any other computer attached to the Internet **100** utilizing a protocol to query the remote vPOP system and obtain capture information, payment administration information, inventory control information, audit information and process payor satisfaction information. Thus, without having to incur the overhead of maintaining sufficient computer processing power to support the vPOP software, a payee can obtain the information necessary to run a husiness smoothly and avoid hiring IS personnel to maintain the vPOP system.

A novel feature of the vPOP software provides payment page customization based on a user's preferences. This feature automatically lists cards that are held by the user and accepted by particular payees based on the active terminal configuration. Each approved card for a particular payee provides smart card information supported by the payee.

Because the security server **140** must sustain reliable operations and enable graceful evolution, it is designed with some important attributes, including: security, availability, performance, scalability, and manageability.

Site redundancy and location redundancy allows the security server **140** to sustain service through virtually instantaneous recovery from internal failures or external disasters that cause physical damage to the system. Minimum-outage recovery is possible with redundant configurations of important components.

The security server **140** supports connections to a proprietary hank network and supports mirrored disk arrays.

The security server **140** architecture supports location redundancy where a secondary remote system is connected to the primary system via dedicated WAN links for software-driven database duplication.

The security server **140** software architecture, the choice of third-party software components, and the selection of hardware platforms enable the security server **140** to gracefully adapt and evolve to take on new demands in different dimensions.

The encryption and decryption algorithms used in processing the messages require significant computational power. A "security processor" is deployed with the security server **140** to hoost the performance of cryptographic algorithms. The security processor is a networked peripheral device to the security server **140**. It provides cryptographic services suitable for processing, and its services are accessible via calls to software libraries.

Security server **140** statistics about transaction requests (by transaction type) and transaction results (e.g., success, failed due to host, failed due to authentication, etc.) can be determined at any time for a particular time interval by generating a report.

A replay attack at the security server **140** is a request where either: a) the request is stale; the request was received "too late" with respect to the reqtate in the request (this window is specified by a configurable security server policy); b) the request is not stale but the exact Request/Response pair Id has already been seen before in a request and still logged in the security server **140** database.

If the vPOP times-out for any reason, it must retry later using a Request/Response Pair Id that indicates a new attempt. If the Gateway receives a late-response (after vPOP has given up) it simply logs it in the database for that retry attempt (if no retry attempt for the transaction is still outstanding at the host). There is a glare situation where the original response could arrive so late that it could be confused with the response from a currently outstanding retry attempt with the host. This situation is logged and the first response not sent back to vPOP.

Finally, the method and apparatus described above may be adapted to process transactions for medical records, prescriptions, audio-visual files, court documents, and any other sensitive or confidential information.

The preferred embodiment of the invention is described above in the Drawings and Description of Preferred Embodiments. While these descriptions directly describe the above embodiments, it is understood that those skilled in the art may conceive modifications and/or variations to the specific embodiments shown and described herein. Any such modifications or variations that fall within the purview of this description are intended to be included therein as well. Unless specifically noted, it is the intention of the inventor that the words and phrases in the specification and claims he given the ordinary and accustomed meanings to those of ordinary skill in the applicable art(s). The foregoing description of a preferred embodiment and best mode of the invention known to the applicant at the time of filing the application has been presented and is intended for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and many modifications and variations are possible in the light of the above teachings. The embodiment was chosen and described in order to best explain the principles of the invention and its practical application and to enable others skilled in the art to best utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A method for a secure transaction over a multi-computer network comprising the steps of:

a. providing at least two separate computer programs that are designed to communicate with each other over a multi-computer network, each separate computer program resident and runnable on a separate computer of the multi-computer network, at least one of the at least two separate computer programs further being a security server program for receiving and processing the secure transaction and at least one of the at least two separate computer programs further being a customer program;

b. running the security server program on a substantially continuous basis thereby making it available to receive secure transactions;

c. running the customer program on an as needed basis for communicating with the security server program with the customer program across a first communication port;

d. receiving a dynamically assigned port address from the security server program, further, receiving from the

security server program a public set of numbers and a security server intermediate value that was calculated using at least the public set of numbers;

e. switching the customer program to the dynamically assigned port address for further communications with the security server program;

f. having the customer program calculate a customer intermediate value using at least the public set of numbers and a shared final value using at least the customer intermediate value and the security server intermediate value;

g. sending the customer intermediate value to the security server program;

h. having the security server program calculate the shared final value using the customer intermediate value and the security server intermediate value;

i. having both the security server program and the customer program create an encryption key using at least the shared final value;

j. having the customer computer encrypt transaction information using the encryption key;

k. sending the encrypted transaction information to the security server program;

l. having the security server program de-crypt the encrypted transaction information; and

m. having the security server program process the transaction.

2. The method according to claim **1** wherein the public set of numbers is at least a public prime number and a prime modulus number.

3. The method according to claim **2** wherein the customer intermediate value is further calculated using a customer selected random number and the security server intermediate value is calculated using a security server selected random number.

4. The method according to claim **3** wherein the shared final value is calculated by the customer computer program using at least the security server intermediate value, the customer selected random number, and the prime modulus; and the shared final value is calculated by the security server program using at least the customer intermediate value, the security server selected random number, and the prime modulus.

5. The method according to claim **4** wherein the step of creating an encryption key using at least the shared final value comprises at least the step of passing at least a portion of the shared final value through a further encryption algorithm.

6. The method according to claim **5** wherein the further encryption algorithm is a one-way function.

7. The method according the claim **6** further including the step of having the customer computer program send customer profile information to the security server program for comparison with customer profile information previously stored on a computer memory accessihly hy the security server program, therehy verifying the identity of the customer.

8. The method according the claim **1** further including the step of having the customer computer program send customer profile information to the security server program for comparison with customer profile information previously stored on a computer memory accessihly hy the security server program, therehy verifying the identity of the customer.

9. The method according to claim **7** wherein the customer profile information comprises a pass phrase that may have white spaces and answers to customer created personal information questions.

10. The method according to claim **8** wherein the customer profile information comprises a pass phrase that may have white spaces and answers to customer created personal information questions.

* * * * *

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Inglis	Art Group:	2768
Serial No.:	09/540,193	Examiner:	Weisberger, Richard D.
Filed:	March 30, 2000		
For:	System, method, and article of manufacture for secure transactions utilizing a computer network		
Atty. Docket No.:	515-001		

Assistant Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Change of Correspondence Address

Dear Sir:

Please change the correspondence address in the above matter to:

Kristofer E. Halvorson
The Halvorson Law Firm, P.C.
1757 E. Baseline Rd. Ste 130
Gilbert, Arizona 85233
(480) 892-2037

Respectfully submitted,

Date: 11/09/04

Kristofer Halvorson, Reg. No. 39,211
The Halvorson Law Firm, P.C.
Attorneys for Applicant
1757 E. Baseline Rd. Ste 130
Gilbert, Arizona 85233
(480) 892-2037